(NASA-CR-187962) KSC-NASA/UCF COOPERATIVE AGREEMENT RESEARCH PROJECTS Annual Status Reports, Jan. - Dec. 1990 (University of Central Florida) 486 p

N91-70698
--THRU--
N91-70706
Unclas
00/61 0333520

KSC-NASA/ UCF COOPERATIVE AGREEMENT

END OF THE FIRST YEAR STATUS REPORT
January-December,1990

UNIVERSITY OF CENTRAL FLORIDA

DIVISION OF SPONSORED RESEARCH

(407) 823-3778 ORLANDO, FLORIDA 32816-0150 FAX (407) 823-3299

March 11, 1991

Mr. Tom Davis
PT-AST
National Aeronautics and Space Administration
John F. Kennedy Space Center
Kennedy Space Center, FL 32899

RE: Annual Report
NASA Cooperative Agreement NCC 10-0003

Dear Mr. Davis:

Enclosed are two copies of the Annual Status Report submitted in fulfillment of the requirement for NASA Cooperative Agreement NCC 10-0003 for the following research projects:

NASA - Project 1 - Intelligent Interactive Visual DBMS
Dr. Ragusa

NASA - Project 2A - Nasa Public Affairs - A
Dr. Driscoll

NASA - Project 2B - Memory-Based Reasoning for Advanced Launch Operations
Dr. Myler

NASA - Project 3 - Natural Language Knowledge Acquisition System
Dr. Gomez, Dr. Segami

NASA - Project 4 - Nondestructive Analysis and Development
Dr. Moslehy

NASA - Project 5 - Materials Performance and Corrosion Protection
Dr. Desai

NASA - Project 6 - Productivity Improvement Techniques
Dr. Hosni

NASA - Project 7 - Project Management
Dr. Rice

NASA - Project 8 - Heat Pipes and Phase-Change Materials for Commerical and Residential Applications
Dr. Gunnerson

The original (reproducible) reports have been sent to the NASA Scientific & Technical Information Facility per instructions in Article 7, NASA Agency Specific Requirements of the Federal Demonstration Project.

If you have any questions concerning these reports, or need any further information, please call me at the above listed number.

Sincerely,

Betsy L. Gray
Contract Coordinator

CC: Joyce Beeson

51-82
333521
P-94

N91-70699

Shuttle Close-Out Photography

Intelligent Interactive Visual Database Management System

Annual Report

Dr. James Ragusa, P.I.
College of Business
Department of Management
University of Central Florida

Dr. Gary Orwig, Co.I.
College of Education
Instructional Systems Program
University of Central Florida

Mr. Gholam A. Shaykhian, Co.I.
College of Engineering
Dept. of Engineering Technology
University of Central Florida

Jan. 25, 1991

Edited by Gary Orwig

Table of Contents

I. Introduction

This research activity began with a draft proposal that was part of the document A proposal to NASA/Kennedy Space Center for a research grant entitled 'Cooperative Agreement for Technology Development'. This composite proposal was submitted on April 5, 1989.

The initial proposal resulted in funding that was provided during the 1989 Summer Term for faculty to research, prepare and present final versions of their proposals. This funding was later extended in order to align the projects with the NASA funding cycle.

The first fully funded year of this research project began in December, 1989. Because this is the first annual report on the project, it will cover activities that occurred both during the preliminary funding and during the first fiscal year.

Research Faculty

Dr. James Ragusa

Dr. James Ragusa is an Associate Professor in UCF's College of Business. Dr. Ragusa serves as Principal Investigator for the NASA/UCF Intelligent Interactive Visual DBMS Project. Prior to joining UCF in January 1987, Dr. Ragusa was employed by NASA's John F. Kennedy Space Center for a period of 23 years. While assigned to the Director of Cargo Operations, Dr. Ragusa was responsible for Artificial Intelligence (experts systems) project activities for payload processing. In addition to overall project and research responsibilities, Dr. Ragusa teaches graduate and undergraduate (honors) expert systems courses. Dr. Ragusa has written extensively and is Manager for the College of Business Intelligent Multimedia Applications Laboratory (IMAL).

Dr. Gary Orwig

Dr. Orwig is an Associate Professor of Instructional Systems in the College of Education at UCF. Dr. Orwig is highly experienced in multimedia database management systems and serves as a frequent consultant to business and industry in system integration of new hardware and software technologies related to computer driven multimedia applications. As a Co-investigator in this project, Dr. Orwig conducts research and provides advice on the procurement of multimedia systems. He conducts research into image oriented expert systems, and serves as a proposal and report writer for the project team.

Mr. Gholan A. Shakhyian

Mr. Shakhyian is an Assistant Professor of Engineering Technology in UCF's College of Engineering. As a Co-investigator he is responsible for object oriented programming, the development of database management systems, and system integration with NASA's SPDMS II. In addition to his faculty and research responsibilities, Ali is a consultant for several NASA contractors.

Research Associates

David Blacklock completed one career as a police sergeant with the Orlando Police Department with his retirement in January of 1990. David began work as a Research Associate in April 1990. He received a Bachelor of Arts from Rollins College in 1973 and a Masters of Business Administration from the University of Central Florida in 1989. He completed the UCF College of Business expert systems course MAN 6938 in the Fall 1988 semester. David has been charged with the following long term assignments in the IMAL: (1) take the lead for expertise in optical technologies, (2) maintain expertise in retrieval technologies, (3) develop proficiency in presentation graphics, (4) develop neural network expertise, and (5) research literature for items of interest to IMAL activities.

Todd Schieffelin graduated from the University of Central Florida with a MBA and undergraduate degrees in Finance and Economics in 1989, 1988 and 1987, respectively. While enrolled in graduate school he worked part-time as a computer consultant specializing in financial analysis via spreadsheets and presentation graphics. While in the MBA program at UCF he enrolled in Dr. Ragusa's expert system class. In this class he met fellow researchers Ann Baron and Dave Blacklock. At the conclusion of the course he worked as an independent study student with Ann Baron, Mike Vine and Dr. Ragusa on the IST grant to interface expert systems technology with laser optics. The result was the "The Chateaux Connoisseur". While not working on the NASA grant, Todd works as a management consultant for a regional insurance company.

Russ Wielgos is currently a graduate student in Business Administration. His undergraduate degree is in computer science from Northern Illinois University, and he has worked as a computer programmer/systems analyst for several years at Arthur Andersen & Co. Russ has been involved in a number of the NASA grant activities, and is currently in charge of the IMAL expert system development and research.

Description of Research Objectives

The original research objectives for the first year of this project follow:

1. Conduct a systems analysis of a large scale traditional image processing operation.
2. Collect data on critical human and performance factors in the areas of image creation, processing, storage, and retrieval. Sample factors include but are not limited to: A) maximum data gathering equipment size allowed, B) resolution required, C) contrast, shading, and color factors required, D) indexed fields - current and future needs, E) desired storage and retrieval strategies, including nonverbal (image based) techniques, F) maximum acceptable retrieval time.
3. Validate the attributes identified above through the creation and testing of analog prototyping models.

The following operational objectives have evolved over the first year of the project.

1. Determine the technological feasibility of intelligent interfacing between AI knowledge-bases information systems and mass storage laser-optical devices for purposes of visual data collection, storage, search and retrieval, and transmission.
2. Recommend improvements to the existing SPC close-out and problems statusing photographic management systems.
3. Ensure that development technology is compatible with SPDMS II and supports LSOC's goal of paperless management of data recording, retrieval, and analysis.
4. Identify other application which could benefit from the resultant technology.
5. Business, Industry Visits, and Workshops

II. Activities and Accomplishments

1. KSC Visits, Contacts

There have been numerous visits to KSC over the contract period. They fall into various categories including: (1) establishing NASA and LSOC technical contacts; (2) acquiring technical and administrative information; (3) briefing NASA, LSOC, and other KSC management representatives. More specifically, project team members (faculty and students) traveled to KSC to work with LSOC shuttle closeout and SPDMS II processing representatives, coordinating with NASA and TGS photo/engineering personnel, and briefing KSC and NASA-Hq. personnel. In addition, various student groups worked in support of this cooperative agreement interacted with various organizational members.

The primary project contacts for this cooperative agreement:

LSOC	John Seaman (Quality Operations) Kevin Paule (Quality Data Center)
Analex	Gene Cain (LSOC support)
NASA	Astrid Heard (Advanced Projects) Tom Davis (Advanced Projects) Nancy Pope (Photo Support) George Dutt (Photo Support)
TGS	Sherline Turbyville Cathy King

The visits to KSC were critical to the task analysis that allowed project team members to chart the entire shuttle close-out photography collection, processing, classification, storage, and retrieval sequence.

2. Survey

During the fall of 1990 a telephone survey was conducted of selected NASA employees and related contractor employees. The intent of the survey was to establish an initial base of knowledge regarding the end users and uses of the photographs. The results of the survey are tabulated in Appendix A, while comments that are related to specific questions are reported in Appendix B.

In summary, most of those who were interviewed indicated that they presently work with Shuttle Close-out photographs in the 8 x 10 color format. Over 75% need to retrieve photographs at least once a month after they have been stored at their facilities. Almost all felt they could not get enough information from black and white photographs, and almost all felt the resolution

provided by the enlargements that they work with is sufficient for most situations. All who were interviewed were positive toward an electronic system if it could provide the same image quality now provided through the enlargements.

3. Business, Industry Visits

Team members have been tracking the development of Digital Video Interactive (DVI) technology by Intel and IBM. Although the motion aspects of this emerging technology have drawn the most public attention, the high resolution, highly compressed still image storage capability of this technology is of particular interest to this research project. As a result, Dr. Orwig attended a DVI Developers workshop that was sponsored by Intel at the 1990 Orlando SALT conference in February. The workshop provided a wealth of information on the topic. Some of the most important information related to planned releases of improved hardware and software.

Dave Blacklock attended the Optical Information Systems Conference in September of 1990. It was targeted toward vendors, users, system integrators, and value added resellers of optical equipment. The conference emphasized the conversion of paper documentation to optical storage and retrieval techniques. Compression standards, object oriented interfaces, remote access to images, and optical storage increasing capacities were discussed.

There have been several opportunities for the UCF project team to demonstrate expert systems/laser-optical technologies to interested workshop and conference attendees. They include:

- The Eleventh International Joint Conference on Artificial Intelligence (IJCAI), August 20-25, 1989, Detroit, Michigan.

- The Florida AI Research Symposium (FLAIRS), April 3-6, 1990, Cocoa Beach, Florida.

- Interservice, Industry Training Systems Conference (IITSC), October, 1990, Orlando, Florida

4. Intelligent Multimedia Applications Laboratory

A new facility, the Intelligent Multimedia Applications Laboratory (IMAL), has been established in the College of Business. The primary purposes are to provide an environment for this NASA research project and for the development of applications using artificial intelligence technology and multimedia devices to solve a variety of business problems.

The IMAL Reference Library

The IMAL houses a growing research reference collection available - on a non-interference basis with on-going grant activities - to faculty, students, and others interested in researching AI and multimedia technologies.

The collection consists of the following:

1. AI Papers and Reports. Microfiche copies of academic papers and reports from Carnegie-Mellon, Stanford, MIT, and Yale that document early AI developments.

2. AI Journal Articles and Research Reports. There are currently more that 1200 AI references spanning the fields of management, MIS, production operations, training, personnel, accounting, finance, economics, and marketing.

3. Expert System Prototype Applications. More that 170 reports have been produced that describe business related knowledge-based system prototypes created by UCF graduate students and undergraduate students.

4. Book Collection. Over 100 books covering AI, knowledge-based systems, AI programming languages, database design, presentation graphics, and multimedia have been cataloged.

5. Conference Proceedings. Important conference proceedings on AI, multimedia, and related subjects are included.

6. Periodicals and Magazines. AI, multimedia, and computer related magazines and periodicals have been collected.

7. Video tapes. Twenty-eight (28) video tapes which describe AI, expert systems, and the process of knowledge engineering are housed in the collection.

Computer aided searching of data records is extremely easy to do. The user is presented a fill in the blank type screen. Depending upon which blanks are filled in, the appropriate search will take place. The reference library databases contain abstracts (usually 100 - 450 words) of each article, book, etc. which the user can search.

The reference library also allows for updates from student assignments. This feature allows faculty to hand out a version of the reference library on a diskette to students in selected classes so that they can enter reference information from assignments directly onto the diskette. Once the assignment is completed the information is appended to the main database.

IMAL Hardware and Software Inventory

Hardware

Zenith Laptop ZTV-3040-MO	$ 4,100
Intel 386 Model PSYP302250DOX	$ 7,543
2 - Zenith Model ZCM-1492	$ 1,118
Sony Trinitron DVM-1390	$ 483
Sony Trinitron DVM-1910	$ 860
Pioneer Laservision Player LD-V8000	$ 2,443
Panasonic TQ-3031F	$18,995
JVC Camera Head TK-870U	$ 2,140
Bencher Copy Mate II	$ 200
4 - Optical Discs	$ 1,280

Software

Dr. Halo	$ 10
VP/Expert SQL	$ 348
Level5 Object	$ 500
1st Class HT	$ 2,505
EXSYS-EL	$ 129
Neural Works Explorer	$ 199
Olmsted and Watkins OWL Neural Networks Library	$ 150
Microsoft Quick C	$ 24
Turbo C++	$ 85
Windows 3.0	$ 115
Owl Guide	$ 108

IMAL Usage Records

On August 29, 1990 a visitor log was began in the IMAL to capture the date, name, organization, and phone number of visitors to the IMAL. As of December 7, 1990, there were 150 visits recorded. There were seventy-three (73) business days of the IMAL. Therefore, there have been an average of two visitors the IMAL per business day.

5. Student Involvement

The research faculty and associates of this cooperative agreement have been supported by the graduate students from Dr. Ragusa's Expert Systems classes and graduate/undergraduate students from the Colleges of Business, Engineering, and Education. The following are the expert system prototype developments that have utilized student talents:

The Chateaux Connoisseur (PC Easy Version)

The original version of the Chateaux Connoisseur was developed under a grant from the Institute for Simulation and Training.

This project represented one of the first efforts to link expert systems to interactive video devices. The video disc that was used as a visual database for the project was a commercial product titled "Salamandre, Chateaux of the Loire." This demo program was expanded to retrieve parameter values from a single database file. This represented the utilization of database retrieval techniques by an expert system. The effort revealed a limited ability of PC Easy to access information from a standard .dbf file. Specifically, PC Easy can only find exact matches. There are no provisions for character string matches or case insensitivity.

The Chateaux Connoisseur (Clipper Version)

The next version of the original demo program was written in Clipper. This represented an opposing approach to that of using a shell program. The Clipper version showed the almost limitless possibilities of using a language, but also the drawbacks of having to start from scratch. First, the development time was increased three-fold. This dealt primarily with building the user interface or menu system. Second, the addition of more "rules" is not as simple of a process as compared to the shell programs. Although the Clipper version contains If - Then statements, clever programming is needed to ensure the "rules" fire in the right order, a process not needed in shell development.

On the plus side, Clipper allowed for much greater flexibility and speed. The flexibility allowed a much more attractive interface with colorful windows. The speed of the Clipper was roughly tenfold. This allowed the menus to change instantaneously after a response was selected, a greatly added benefit.

The Chateaux Connoisseur (Guide Version)

The Guide version represents a move into the object-oriented world. Guide is an authoring system that runs under Windows 3.0. The Guide version was not successfully completed. A snag arose when researchers tried to step forward and step backward within a predetermined range. A remedy to the problem would be to write a laser disc driver that would be Windows and Guide compatible.

The effort resulted in several telephone conversations with Owl International, makers of Guide. Most of the conversations were of little or no help. In fact this project appears to be the first to interface their product with a laser disc player. The Guide version did have a distinct advantage over the other versions -- the hypertext interface. This allowed the user to move quickly within the information in a format he/she chose.

The Generic Database Program

The generic database program was developed replace the need for Ashton Tate's dBase III. The generic database program allows a user to create a database file from scratch. Additionally, the program allows a user to edit, add and delete fields from an existing database. Further, a provision was designed to add, edit and delete records to the database. The thought behind the program was to allow a user to create a database file and design an expert system front end to classify (populate) and retrieve (search) data.

The code for generating all the panels (screen menus) is developed in Clipper database language. In addition, the communications between the expert shell and the laser disc are routed through the Clipper codes.

The player and/or recorder part of the laser disc is controlled by the code written in C++ language. The CPU and the analog laser disc are connected via a serial connection (standard RS-232). The laser disc receives a frame number from the CPU for display purpose and displays the image located at that frame number. The collected visual data and the frame number read by C++ are appended to the database and the image is stored on the laser disc.

The Database - C Driver (Pioneer Disc Version)

The Pioneer version of the Database - C Driver allows a user to search a database and to control the Pioneer laser disc player from the DOS prompt. The Database - C Driver will conduct a search of up to ten fields of a named database. If a match is found it will display the appropriate images. A special feature of this driver is that it allows for the stepping forward and backward through matching records. The thought behind this driver is that it allows one to write his or her own searches. This allows one to use indexes and it addresses the previous problem of limited character string search capabilities. A major problem with this driver is that it requires 160K. This presents a problem when trying to use it with many of the shell programs.

The Database - C Driver (Panasonic Version)

The Panasonic version is essentially the same as the Pioneer version. The only difference is that it controls the Panasonic Optical Disc Recorder/Player.

NAPSAC and PRAISE

During the Fall semester of 1989, two student groups from Dr. Ragusa's Expert Systems class visited the Kennedy Space Center, TGS and Lockheed to gather necessary information for building

both photo classifier and photo retrieval expert system prototypes. These expert systems would address the problem of processing and storing the multitude of space shuttle pictures taken for each launch.

Following various knowledge engineering sessions, both student groups built their respective prototypes, which they called NAPSAC (NASA Photo System to Aid in Classification) and PRAISE (Photo Retrieval And Intelligent System Expert). A demonstration of these prototypes, along with an executive summary report, were presented to NASA, TGS and Lockheed officials.

These original prototypes were built using the Texas Instruments' PC EASY product and, although the shell proved quite capable of handling the requirements of the two applications, the project groups noticed many drawbacks in using this particular software package. These drawbacks prompted a search for a more capable shell for use with these two applications. The resulting research consisted of new product education, as well as the actual cloning of the original two applications in various other expert system shell software products. The hands-on results of this continuing research are provided in Appendix C.

Expert System Selection Advisor

Because a large number of expert system shells have been examined in relation to this project, an expert system is under development that will help advise a user on the appropriate expert system shell to user for any specific application. This shell will capitalize upon the experiences of the faculty and students who have had direct experience with various shells.

III. Budget Projected vs. Real Summary (Nov 1, 1989 to Dec 21, 1990)

	Planned	Actual
Faculty Salaries	$59,253	$60,493
Students/Associates (OPS)	$30,520	$40,746
Expenses	$18,500	$12,046
Equipment(OCO)	$43,356	$40,296
Overhead	$53,054	$53,184
	-------	------
Totals	$204,683	$207,062

As can be seen above, the planned vs. actual expenditures are very close to each other. Specifically, very little variance took place in faculty salaries. The greatest variance was in the category of student/research associate expenses. The difference of $10,000+ was absorbed from expenses and equipment. It is felt that these increased expenses were in the best interest of the long term goals of the project.

IV. Preview of Next Year

The plans for the 1991 project year are directed toward a transition from analog image storage and retrieval toward digital image storage and retrieval. The use of expert systems to drive digital image database classification, storage, and retrieval systems will be examined. Due to the high resolutions required for the digital images, it is anticipated that image compression /decompression software will be required. Finally, it is anticipated that interface drivers will need to be written to link the expert systems with the image capture, compression, and decompression software.

Elements of interest in this research relate to file sizes, compression ratios, quality of images once they are decompressed, and retrieval/decompression access times. While some performance information is available from individual vendors, virtually no performance data has been reported for integrated systems. This systems integration research will help to determine the parameters under which digital image storage and retrieval systems might be implemented.

In relation to this expansion into digital technology, it is anticipated that several purchases will be required. In particular, a color scanner, a video frame grabber, a 1280 x 1024 pixel display system, and image processing software have been specified.

While DVI and competing technologies will continue to be tracked, they have been removed from the purchase list for the short term. At the present these technologies are still emerging, and as a result there are few commercial software packages available that will serve the needs of this research.

Most of the research of the previous year will continue. In particular, additional expert system shells will be evaluated. Also, the Level5 Object software will be examined in greater detail to determine its strengths and weaknesses in accessing image databases. Interfacing techniques will continue through object oriented programming. In particular, C++ has been chosen as the high level language for the device drivers and other interfaces that will be required.

Finally, research will continue into intelligent object oriented database management systems. Specifications for several commercial products now indicate a combination of the expert system and database management functions into one software system. Because these products are in an early stage of product evolution, there exists a need to determine how well they can perform in prototype applications.

V. Appendices

Contents

Appendix A

Shuttle Close-Out Photography
End User Survey

UCF - NASA Project
Shuttle Close-Out Photography
End-User Survey

Survey Summary

A. User Profile

1. Employer's Name LSO, TU-MSD, USBI, Morton Thiokol, MMC, USBI

2. Location (City, State) KSC, Brigham City, New Orleans, Huntsville

3. Job Title Almost all engineering, management

4. Very briefly state your job responsibilities as they relate to the shuttle.
   Quality Control, Engineering Review, Public Relations

B. Current / Potential Need for Closeout Photographs

5. Do you currently make use of Shuttle Close-Out photographs?

   YES 100% NO ______

   If NO, go to Question 13
   If YES, please answer through number 12.

6. What size of prints do you prefer to use? (Check one or more.)

   8% Contact Sheets
   92% 8 X 10 Prints (23% use both)
   _____ Prints larger than 8 X 10
   _____ Other (Briefly describe)

7. Are the prints delivered soon enough for them to be of maximum value to you?

   YES 69% NO 31%

8. How are the prints stored and accessed at your facility?

   _____ We throw them out when we are done with them.
   92% We file them in cabinets.
   _____ We have a computer-based logging system.
   8% Other (Briefly explain) (Boxes)

9. How often do you need to retrieve or replace a photograph that has been stored or thrown out?

31% More than once a week
15% Once a week
31% Once a month
23% Once a year

10. Could you get by with black and white pictures?

YES 8% NO 92%

11. Is the **quality** of the color of a photograph important to your use of it?

YES 77% NO 23%

12. Do most of the close-out photographs contain sufficient image resolution? (Can you see what you need to see?)

8% No, I need more image resolution (clarity).
92% Yes, they have sufficient image resolution.
_____ Yes, in fact, they have **more** resolution than I need

13. If color and detail were similar to what you use now, would you consider working with an on-line (computer) image retrieval and delivery system?

_____ No Way
92% Sure
8% Maybe, if it would do the following:

(No specific comment regarding the "maybe")

The following are questions for those who don't currently use Shuttle Close-Out photographs.

14. I don't currently use these photographs because:

_____ They haven't been made available to me.
_____ They aren't needed for what I do.
_____ Other (Please Explain)

15. If an on-line (computer) image retrieval system were available to you for these photographs, ...

_____ I wouldn't need it.
_____ It might be useful in the following way(s) ... ______________________________

____

Appendix B

Shuttle Close-Out Photography
NASA Survey Comments

NASA Survey Comments

5. Do you currently make use of Shuttle Close-Out photographs?

(YES)

For flight readiness reviews and mandatory review requirements

For ET/SRB groups (10 engineers review each photo)
ET approx. 400 photos per mission
SRB approx. 400 photos per mission

For Q.A. function, historical documentation, anomaly investigations

6. What size prints do you prefer to use?

(Contact Sheets)

Too small, but sometimes used for identification purposes

Too small, not enough resolution when used with a magnifier

(8 x 10 Prints)

Expensive, use only when necessary

7. Are the prints delivered soon enough ...?

(YES)

Normally takes 2.5 to 3 weeks
If emergency, can sometimes get them the next day

Usually available in 2 days

Usually available in one week

We don't usually need them right away

(NO)

14 - 15 days are normal

Usually two week turnaround
Would prefer normal next day turnaround
A good reason for video would be to see if the pictures are ok initially

8. How are the prints stored and accessed ...?

(We file them in cabinets)

Manual storage by photo number
Keep photos for 1 year, then put into long term storage
Arranged by area (takes about 10 minutes to find)

Filed by topic, procedure, and number

Filed by subject, area, and date

Filed by topic

Filed by flight

Filed by categories
Filed by engineers
Organization
  ET: 50 groups
  SRB: 25 - 30 groups
Would like to be able to review based upon date/subjects rather than memory
Photos older that 10-12 vehicles now in long term storage

Filed by mission/subject, but with no consistency

Filed by launch

9. How often do you need to retrieve or replace a photograph that has been stored or thrown out?

NOTE: I think this question was interpreted two ways. It appears that some of those who didn't check one of the options or who picked "once a year" were thinking "How often do you need to access any ONE PARTICULAR photograph?" while others understood the question to mean "How often do you need to retrieve ANY photograph?"

(More than once a week)

Approximately 200 retrievals a year

We must make daily retrievals

(Other Comments)

On occasion

Variable

Rarely

Once a year or less. There are some photos that we may not ever need to retrieve.

10. Could you get by with black and white pictures?

(NO)

Color very important, Could not do without. Eg. Bonding, primer

Looking for discoloration
Some tags/pieces are color coded. Eg. Red - Remove for flight

Color absolutely essential

11. Is the quality of a color photograph important ...?

(YES)

Overall, photographers do a good job of maintaining quality

Operator skills are important

(NO)

Shade variations not always critical

Shade of color not critical, can mentally adjust

12. Do most close-out photos contain sufficient image resolution?

(YES)

Depends on lighting/aperture. Present lighting a problem

In 80% of the cases, ok. Would like more resolution

contact prints not good enough

Ok in most cases, but sometimes not enough

OK, but cannot be less and still be acceptable

13. Would you consider on-line ... ?

(YES)

Need hard-copy, slides, color output

Would love it for review

Must have hard copy 50% of the time

Want terminal for faster retrieval

Would be good for static storage

Appendix C

Comparison of Expert System Shell Software

COMPARISON OF EXPERT SYSTEM SHELL SOFTWARE

Over the past year, the NASA Photo Classifier (NAPSAC) and Retriever (PRAISE) expert systems have evolved from a pair of large, slow moving student projects into two more efficient expert system prototypes. This evolution was made possible through explorations with several expert system shells. The following is a summary of the advantages and disadvantages of various expert system products in relation to these two applications.

PCEASY Advantages

- The use of color in the various title, question and conclusion screens make the system more appealing to the eye and quite user-friendly.
- The length of the inputed photo descriptions are only limited by the size of the data base field it will be written to.
- The length of the various user Help screens is limitless.
- The finished knowledge base can be converted to 'C' language to increase runtime speed.
- All application functions, such as Help screens and description inputs, are internal to the PC Easy software. The only external interface needed is with dbase.
- There is no real limitation on parameter names and lengths.
- Conclusions can be presented to the user on one screen.

PC EASY Disadvantages

- The unfinished application has already reached its memory constraints.
- The runtime and development speed are quite slow.
- The initialization questions, such as date of photo, photographer, KSC number, etc., each needs to be handled using one question and response per screen (a total of 10 screens).
- PC EASY has an inability to hold large rules without impeding response and development time.
- There is an absence of on-screen validation techniques.
- The shell requires the application to send a value for each data base field in each record, regardless of whether the field contains a value or null.

- PC EASY returns very non-descriptive error messages.

KAPPA Advantages

- Initialization prompts, such as date of photo, photographer, KSC number, can all be inputed using one input screen.
- Rules can be formed using menu screens and/or written text-type screens.
- The shell runs under the user-friendly Microsoft Windows package.
- The shell gives the application the user of object-oriented programming. At the present time, this feature is not crucial to the application.

KAPPA Disadvantages

- The system response time is quite slow.
- There is a limit of 190 characters on the user help screens.
- Input description fields are limited to 31 characters.
- When a rule is not satisfied, the system will ask the user for the value of the goal. This is not always required.
- Conclusion screens cannot easily be shown using one screen.
- No color choice of text is available.
- There may be a bug in Kappa Version 1.0, due to the fact that some rules are not fired upon correct answering of the questions. Identical inputing of these fields may lead to a different result (i.e. correct firing of the rules).
- The purchase of Microsoft Windows is an extra cost.
- The Kappa manual is not very well written or complete and fails to describe the definition of object-oriented programming very well.

1st-Class Hypertext Advantages

- Rules can be induced by filling in a spreadsheet-type form.
- Rules can be presented in an easy to read structure tree.
- The shell has the capability of using hypertext for help and conclusion screens.

- The numeric initialization fields, such as date of photo, roll number, and frame number, can be input on one input form.

- Color may be added to textual questions and help screens.

- Conclusions can be presented to the user on one screen.

1st-Class Hypertext Disadvantages

- There is a database limit of 65,000 records.

- Large applications are difficult to put into example form. These examples are used to induce rules.

- There is no capability for textual input, such as the photographer and photo description.

- Parameter names are limited to approximately 8 characters. This causes multiple choice answers to be abbreviated to the user.

- There is no validation of numeric input ranges.

- No confidence levels are allowed on rules. This feature was not crucial to this application.

- The length of a rule is limited to approximately 840 lines. This disadvantage can be solved by linking various rules (i.e. knowledge bases), but this linking will slow down consultation time.

- The shell only allows for 32 factors (parameters) and 128 values for each factor.

EXSYS EL Advantages

- Ability to create own screens and windows

- Input maximum is 1,000 characters

- Ability to quickly rerun the consultation with new input choices

- Three types of confidence level use

- Easy to Learn and Use (possibly easier than PC Easy)

- Written in C language making it quicker than PC Easy

- Has rules and text in two different files. This also will aid speed.

- DBASE and Lotus Interface
- External Linkage possible
- Ability to generate report with Report Generator
- Complete, easy to read manual

EXSYS EL Disadvantages

- No Form creating capability
- No mouse capability
- Creating a knowledge base requires manipulating around a number of menu screens. No rule text editor is available. EXSYS Professional has the capability to take rules from Word Perfect and compile them into KB files.
- No hypertext feature for help screens. This is available in EXSYS Professional.
- Limited number of runtime installations
- Knowledge base name must be known in order to access it. There is no menu containing KB names to choose from.

VP-EXPERT/SQL Advantages

- Ability to create input forms for users to enter data
- Mouse capability
- Provides a runtime and development mode
- Good search functions
- Fast consultation time
- Allows developer to type rules into a textual editor
- Very easy to learn and use
- Capability to induce rules from tables
- Contains hypertext capabilities (not tested by author)
- Dbase, SQL and LOTUS interfaces
- Ability for external linkage
- Contains a graphics text tree for tracing knowledge flow

- The created NAPSAC and PRAISE knowledge bases take up a tenth of the space as they do in PC Easy

VP-EXPERT/SQL Disadvantages

- Input length is limited to 30 characters
- Field names are limited to 20 characters
- No user help is provided with completed expert systems
- Unable to find any graphics capabilities
- Programming of the input form can get quite complicated
- No input validation
- The user needs a mouse AND the Return key to manipulate around the input form

At the present time, it has been determined that the most suitable expert system shells for the two prototype applications (NAPSAC and PRAISE) to be VP-Expert/SQL and 1st-Class Hypertext, in that order. The original applications have been recreated and expanded upon using VP-Expert/SQL, The results are much quicker applications, using only one-tenth of the memory space as PC-EASY. The improved prototypes are also more efficient and user friendly.

During the second year of the project, a review will be conducted of various object-oriented expert systems shells, such as Level 5 Object and KBMS, to see what value they may lend to the NAPSAC and PRAISE applications. The review of non-object-oriented shells will also be continued. This hands-on approach, should allow the faculty and students to become better informed on what these various shells have to offer.

Appendix D

Software Interfaces

Software Interfaces

Laser Disc Recorder/Player Drivers

The general purpose of the following programs is to provide a user-friendly interface between an artificial intelligence expert system and a laserdisc recorder/player through the use of DBase files and C++ programming.

There are three sub-purposes as follows: to provide a user interface, to provide a DBase file interface, and to provide communications between a laserdisc Recorder/Player and an IBM PC.

These sub-purposes are supported by three sets of programs. The programs will be discussed in terms of the sets to which they belong.

Functional Descriptions of Programs

User Interface

The user interface programs are listed below. All of the programs are functional and complete. However, all of the programs could be enhanced to take advantage of the Windows environment.

BUNCH. H
COMMON.H
DISPLAY.H
GPOPUP.H
HEADER.H
LIST.H
MNUM.H
CAL.H
EVERT.H
GTEXT.H
MANAGER.H
MOUSE.H
OBJECT.H
PEN.H

DBase III Interface

Only one program supports the DBase III interface - DBASE.H. It is considered incomplete. It should have the following enhancements: (1) DBase searching for multiple indexes and (2) the present limit of 50 records should be eliminated using dynamic memory management.

Communications

Only one program supports the Communication interface - SERPORT.H. It is considered incomplete. It should have the following enhancements: Detailed Serial Port error report.

Major Functions of Programs

User Interface

The major function of each program is as follows:

BUNCH. H	-	Generates the windows environment.
COMMON.H	-	Provide a collection of all the user interface variables.
DISPLAY.H	-	Display the current text value
GPOPUP.H	-	Provides a pop-up Window.
HEADER.H	-	Provides a structured data type.
LIST.H	-	Lists the data structure.
MNUM.H	-	Generates window for search by number.
CAL.H	-	Generates a menu window.
EVERT.H	-	Provides a window data type.
GTEXT.H	-	Provides a text input/output function in a graphics environment.
MANAGER.H	-	Manages window functions.
MOUSE.H	-	Supports the mouse.
OBJECT.H	-	Provides a window object function.
PEN.H	-	Provides a set of primitive graphic functions such as a square, circle, and a line.

DBase III Interface

DBASE.H has the function of searching picture information from the DBase files.

Communications

SERPORT.H has the function of setting communication modes and sending or receiving commands between the laserdisc recorder/player and the computer.

Compiling Information

Project file is used to generate a compiled program. Project program contains a set of files which include all source codes. The file extension of project file is .prj.

Disk.prj - generate a integrated program.

Laser.prj - generate a Pioneer interface program.
New.prj - generate a Panasonic interface program.

Disk.prj

BUNCH.CPP: Class Bunch allows the organization of Panes into any type of grouping desired. Bunch is a subclass of Pane. Bunch has a subclss, Manager.

Header Filels: BUNCH.H
COMMON.H

COMMON.CPP: Collection of commonly used functions and macro definitions

Header Files: COMMON.H

DISPLAY.CPP: Display depicts the window where the current value is display. As a subclass of Pane Display inherits methods and data objects from Pane and some methods from Bunch. The Display uses its constructor to clear the display and setup the Pens to paint the display. Class Display has seven public methods which have been redefined in this area class to meet Display need.

Header FIles: BUNCH.H
PANE.H

GPOPUP.CPP

HEADER.CPP

LIST.CPP: Class List performs the role of a general list manager and a list iterator. Class List consists of three class: l_node, List, and ListIterator. Class l_node is a linked list node and used by the general data structure classes, List.

Header FIles: List.h

MNUM.CPP

CAL.CPP

EVENT.CPP: Class Event performs a housekeeping job while working in conjunction with class Manager and Pane. Class Event records the type of mouse that has occurred.

Header Files: Event.h
Mouse.h

GTEXT.CPP

MANAGER.CPP: Class Manager performs the task of general management for the graphical control interface environment. To be displayed on the screen, a pane must be inserted ether directly or indirectly into Manager. Class Manager is a subclass of Bunch.

HeadFiles: MANAGE.H

MOUSE.CPP: Mouse Function implementation.

HeadFiles: Pen.h

OBJECT.CPP Objects on the screen like icons, menu selections, and other graphics figures can be put on a list and selected using the mouse. Each object has a function associated with it that is executed when the object is selected.

HeadFiles: <alloc.h>
<graphics.h>
<conio.h>
"mouse.h"
"object.h"

PEN.CPP: Class Pen is designed to control the pen status and movements on the screen, such as color, position, or mode.

HeadFiles: "Pen.h"
"Pane.h"
<Graphics.h>

PANE.CPP: Class Pane provides a restricted area for Pen to operate. Pen is allowed accesses to Pane's private data and member functions to draw icons or write text by declaring.

HeadFIles: "Pen.h"
"Pane.h"
<Graphics.h>

SERIAL PORT FUNCTION

The serial port library has several functions used to facilitate IBM PC serial port. Using these function, user easily create the customized laser disk control commands. These functions will work on Pioneer and Panasonic laser disk player. These serial port functions allow you to:

1. Initialize communication mode.
   - Communication port
   - Baud Rate
   - Parity
   - Character bit
   - Stop bit
2. Send character or string to disk player.
3. Receive character or string form disk player.
4. Contains Panasonic laser disk command.

Function Description

Name: serial(int, char*, unsigned char)
Header: SERPORT.H
Type: class
Purpose: contains a port initialization function.
Parameters: port number(0: for com1, 1: for com2)
port name : any string
initialization code: Baud_rate || Stop_bit || Data_Bits

* Note: See Serport.H for more detailed parameter options

Name: send(char*)
Header: SERPORT.H
Type: public
Purpose: send a character string to port
Parameters: character
Return: None

Name: Receive(char*)
Header: SERPORT.H
Type: public
Purpose: receive a character string from port
Parameters: character
Return: None

Panasonic Command

friend void fwd_play();
Purpose: forward play command
Parameters: None
Return: None

friend void rev_play();
Purpose: backward play command

```
   Parameters: None
   Return:     None

friend void fwd_step();
   Purpose: forward step command
   Parameters: None
   Return:     None

friend void rev_step();
   Purpose: backward play command
   Parameters: None
   Return:     None

friend void fwd_scan();
   Purpose: forward scan command
   Parameters: None
   Return:     None

friend void rev_scan();
   Purpose: backward scan command
   Parameters: None
   Return:     None

friend clear_command();
   Purpose: clear command buffer
   Parameters: None
   Return:     None

friend * read_frame();
   Purpose: read frame number from laser disk player
   Parameters: None
   Return:     None

void command_type0(char cmd[2]);
void command_type1(char cmd[2], char num[]);
void command_type2(char cmd[2], char num1[], char num2[]);
   Purpose: create Panasonic control commands as shown in manual,
            page 57-58.
```

Mouse Functions.

The mouse library has several functions used to facilitate Microsoft compatible mice. These functions will work on Microsoft mice or other mice using a Microsoft compatible driver. These mouse functions allow you to:

1. Initialize mouse/determine if mouse exists.
2. Get the status of button presses/releases.
3. Hide/reveal the mouse cursor.
4. Get/set the mouse cursor position.
5. Select type of mouse cursor (hardware or software).
6. Adjust the mouse sensitivity (speed).
7. Get information on direction of mouse movement.
8. Establish horizontal/vertical boundaries of mouse

movement.

Pen Functions

Class Pen has the following private data to indicate its status:

- mode : mode of setfillstyle
- linetype: line type for graphic function(see Turbo C++ graphic library)
- linepattern: line pattern for graphic function(see Turbo C++ graphic library)
- thickness: line thickness for graphic function(see Turbo C++ graphic library)
- color: graphic color(see Turbo C++ graphic library for the number of color information)
- x, y: position of the graphic components

Pen works closely with class Pand. Using its constructor and several member functions to work Pane, Pen can be call to perform:

```
Name: SetLinePattern(int, unsigned, int)
Purpose: Set line pattern(line type, pattern, thickness)
Parameter: 1: line type
           2: line pattern
           3: line thickness
Return:  None

*note: the line setting is based on turbo line drawing option.
```

```
SetMode(int)
     Purpose: Set mode of private parameter with class
     Parameter: 1:  mode of fill pattern(see Turbo C++ Manual)
     Return:  None

MoveTo(Pane *, int, int)
     Purpose: Set position of Text
     Parameter: 1. Pane pointer
                2. X position
                3. Y position
     Return:  None

     * This function is same with gotoxy function except Pane
       Pointer.

Text(Pane *, char *)
     Purpose:  print character string on the screen. The position
               can be set by MoveTo function.
     Parameter: 1. Pane pointer
                2. Charter String
```

```
     Return:    None

Line(Pane *, int, int, int, int)
     Purpose: draw line
     Parameter: 1. Pane Pointer
                2. starting x position
                3. startint y position
                4. ending x position
                5. ending y position
     Return:    None

Rect(Pane *, int, int, int, int)
     Purpose: draw rectangle
     Parameter: 1. Pane Pointer
                2. starting x position
                3. startint y position
                4. ending x position
                5. ending y position
     Return:    None

Rect(Pane *, int, int, int, int)
     Purpose: draw filled rectangle
     Parameter: 1. Pane Pointer
                2. starting x position
                3. startint y position
                4. ending x position
                5. ending y position
     Return:    None
```

Pane Functions

Class Pane has the following private data to indicate its status:

```
     xoff, yoff:  offset from lower left corner of screen
     xrel, yrel:  offset from lower left corner of parent bunch
     xmax, ymax:  Maximal local coordinates
     fg_box clipbox: pane's area on screen
```

```
Header Files:  <graphics.h>
               <stdio.h>
               "list.h"
               "event.h"
               "common.h"
               "gtext.h"
```

Pane works closely with class Pen. Using its constructor and several member functions to work Pen, Pane can be call to perform:

```
virtual void Configure();

virtual void Draw();
```

```
virtual void Handle(Event &);

int Inside(int, int);

     Purpose: Set line pattern(line type, pattern, thickness)
     Parameter: 1: line type
                2: line pattern
                3: line thickness
     Return:  None

     *note: the line setting is based on turbo line drawing option.
```

Class PaneList has the following public function:

```
void Append(Pane *n) {List::Append((void *) n);}

void Insert(Pane *n) {List::Insert((void *)n); }

void Delete(Pane *n) {List::Delete((void *)n); }

void Replace(Pane *n, Pane *r) {List::Replace((void *)n, (void
```

Laserdisc to Computer Interface (Panasonic)

by Hoi Yoo, 9/25/90

This documentation describes a Panasonic laserdisc interface with an IBM PC. The interfacing can be achieved by a RS232 communication prototype controlled by a computer program. The program used is Turbo C++, by Borland International, Inc. Laser disk pictures are coded into a Dbase III file with a picture name, start frame number, end frame, and type of picture. This documentation only includes the technical specification of laserdisc communication.

This documentation for interfacing consists of three parts: connection, communication mode, and control commands.

Connection

There are two connections options. See Panasonic operating instructions in metal file cabinet. On page 50, Examples 1 and 2 show types 1 and 2. In the connection, we use type 1.

Communication Mode

See page 49 of the operating instructions for the "Communication mode set up". Changes to the written set up are as follows:

1.	Baud rate	2400 baud
2.	Character bit	8 bit
3.	Parity	no parity
4.	Stop bit	1
5.	Communication type	Type 1

6. XON/XOFF (have used XON, but does not matter; could be XOFF)

IBM PC communication port : serial port (can choice either port1 or prot2). However, have used port1.

Control Commands

See page 53 - 59 of operating manual for control commands.

Most important control command is the COMMAND FORMAT, page 53., no. 2. Every command must be followed by the command format. More detail is offered on page 56 - 59 for individual command types.

Control command to make a connection between the laserdisc and the computer is as follows:

See page 80, no. 5, for ON-LINE CONTROL COMMANDS. This command must come first with every communication to establish communication with the laserdisc.

Table 1, page 80, shows On-line modes. There are 16 options for modes. We use mode 0.

Appendix E

"Expert systems and imaging: NASA's start-up work in intelligent image management"

Page intentionally left blank

Expert Systems and Imaging: NASA's Start-Up Work in Intelligent Image Management

James M. Ragusa and Gary W. Orwig

An ambitious, three-year cooperative research effort between the National Aeronautics and Space Administration and the University of Central Florida (UCF) is exploring how expert systems and imaging technology (laser optical devices) can be integrated into an intelligent, interactive visual data base management system. If expert systems developers think this could be the start of something big, they'd be right: One of the project's goals is to develop a prototype high-definition television expert system.

This article, the first of two, introduces the problem of image management for NASA's space shuttle program and discusses the role expert systems have in the big picture. Part two will appear in a subsequent issue of EXPERT SYSTEMS.

As many as 100,000 color photographs are taken for every space shuttle prelaunch at Florida's Kennedy Space Center. Most of these pictures, called shuttle close-out photographs, are used to establish a baseline from which to evaluate virtually every step of flight preparation for the space shuttle's orbiter, external tank, and solid rocket boosters. Every crucial flight element, subassembly, cable and hardware connection, and layer of insulation is pictorially documented as it is installed, removed, or replaced. These photographic images are then used by engineers from NASA and its contractor companies at the Kennedy Space Center and other locations to verify that essential installation, test, or repair procedures have been completed.

The post-*Challenger* era has made readiness verification and safety for flight a number one priority; in a sense, close-out photographs provide a second and third set of eyes to independently verify that the final preflight configuration is ready for flight. Earlier flight photographs might help to determine whether a new structural fault developed suddenly over a period of time. There are numerous examples of photographs from earlier space shuttle missions providing answers to questions raised during flight readiness preparations.

On other occasions, a sequence of pictures may be needed during real-time space shuttle processing. For example, if an indicator light isn't working properly, a photograph can confirm whether a critical cable connection had been properly made. In many cases, a photograph can eliminate the need to disassemble many layers of insulation and components to physically verify that a cable connection had been completed days or weeks earlier.

The problem of image management

Not surprisingly, 100,000 pictures per launch, multiplied by dozens of launches, leads to the question of how to access and manage an image data base. In their original form, close-out photos exist as 35-mm color negative film slips with 6 to 8 images per slip. Proof pages contain positives of all the images on each roll of film. In addition, multiple 8-by-10-inch

James M. Ragusa is an associate professor of management at the College of Business Administration, University of Central Florida in Orlando. Formerly, he was employed with NASA at the Kennedy Space Center. He can be reached at (407) 275-2915.

Gary W. Orwig is an associate professor at the College of Education, University of Central Florida. He can be reached at (407) 281-5179.

color enlargements are made of any images that require close inspection.

With the exception of automated film processing, all handling of the images, including the filing system, involves extensive manual labor. Complex problems frequently evolve when engineers attempt to retrieve an individual photo or a group of related pictures because only one true access point is allowed. Without the appropriate accession numbers, pictures cannot be retrieved from the files. Several manual and microcomputer data base management systems assist in the identification of the accession numbers, but even the best of these allow only limited searches based on brief descriptions and key words.

Captions, however, may not fully identify a picture's content and may not reflect evolving terminology used as key words. As a result, the data base does not always provide ready access to needed pictures. In many cases, the success of the retrieval process depends almost entirely on staff members using processing heuristics (i.e., rules of thumb) gained from years of experience. Enter expert systems.

Accessing high-definition images

There is little doubt that expert systems can assist in the collection, classification, storage, and retrieval of high-quality pictorial images that exist in a variety of forms. Images can either be line drawings or photographs. They can be in color or black-and-white, with few or many shades, and may be blurry or sharp. Images may be analog in form, whether they occur as silver and dye on film or as lines of light on a television screen. Of particular interest to the NASA project are digital images, wherein many small picture elements, or pixels, have discrete characteristics that are identified by predefined numeric codes. Computers can process digital images for storage, transmission, and retrieval.

A 35-mm color slide that has been properly exposed and developed is the standard of quality. Although measurements of specific variables such as lines of resolution, color saturation, and maximum light/dark ratio are defined for this medium, for simplicity's sake, any image whose quality is equal to a good slide can be termed a high-definition image.

Text field searches in image data bases

Compared with the advances in alphanumeric expert systems–assisted data base management, remarkably little progress has occurred in the management of high-definition images. Most high-definition image data bases can still be described as manually generated, manually processed, file-cabinet-stored, manually retrieved systems. If alphanumeric information were still treated this way, most modern businesses could not exist. The NASA-sponsored research project is investigating how high-definition images can be processed through a type of expert systems–assisted relational data base management system.

The NASA-sponsored research project is investigating how high-definition images can be processed through an expert systems–assisted relational DBMS.

Elsewhere, too, research is taking place in the design of image and visual data base management systems. K. Meyer-Wegener and others at the Naval Postgraduate School have described a process for image data base management in a multimedia system. These researchers have proposed the use of text description fields in an otherwise traditional alphanumeric DBMS to allow users to indicate the content of images. A structured query language is then implemented to allow users to search the text fields. Software interfaces allow the system to retrieve image files that match queries.

A picture is worth a thousand words

The use of text description fields will be one essential route for classifying and accessing high-definition images. More complicated is the need to access a picture within a picture. For example, an engineer identifies an abrasion problem on a plastic sheathed cable, where it rubs against a metal sheathed cable. Using text searches to locate every occurrence of this situation in an image data base of a machine as complex as the space shuttle would

work only if all appropriate text entries existed. Because any single picture could yield thousands of words of description if this amount of detail were required, the process of describing each picture would be time-consuming, if not impossible, and the resulting text data base would be enormous. A far more logical solution is to locate an example of the problem in an existing image, identify it by drawing a box around it, and then ask the computer to search for any other similar patterns in the entire visual data base.

Image processing and pattern recognition

Although this type of retrieval process is still in development, it does exist. The Query by Pictorial Example (QPE) process allows access to image data bases through image processing and pattern recognition techniques.[1] Using this method, queries can be expressed in terms of pictorial examples.

At present, image processing and pattern recognition of image segments requires advanced computing power. However, even desktop computers can manage less sophisticated image data bases. A prototype image data base system that runs on a standard desktop computer enhanced with image capture boards and write once/read many times (WORM) disk drives is being used to collect information on the next generation of medical information systems. Representatives of the Atlanta Police Department have examined a prototype electronic mug-shot image data base that also uses image capture boards in a standard desktop computer. The Law Enforcement Agency Data System with Pictures is intended to integrate several of the functions associated with collecting, using, and maintaining color mug-shot files.

Intelligent, interactive image management

Advanced image data base management systems might now be practical because the key technologies—standard DBMS systems, expert systems shells, hypermedia software, image digitization, and high-definition video—already exist. In particular, NASA's Intelligent Interactive Visual Data Base Management Systems project that is using space shuttle closeout photography as its case study is targeting an expert systems–driven intelligent management enhancement of image data bases. In addition, the project is identifying the emerging technologies associated with the electronic capture, classification, storage, retrieval, and transmission of high-definition images.

The phase 1 prototype already demonstrates the intelligent retrieval of still images, motion sequences, computer-generated graphics, text, and multiple audio channels.

The project's research objectives can be defined as follows:

- To conduct a systems analysis of a large-scale traditional image processing operation.
- To collect data on critical human and performance factors in the areas of image creation, processing, classification, storage, and retrieval.
- To validate the attributes identified through the creation and testing of prototype models.

Project phases

The project's research areas—systems analysis, image systems, alphanumeric systems, and expert systems—are divided into three phases (see the inset for a complete overview of research objectives). The first phase (year 1), currently under way, is concerned primarily with understanding the dynamics of the existing close-out photography system, developing analog laser disk data bases, and identifying appropriate expert systems technologies. Phase 2 carries over into digital optical systems by evaluating user reactions to prototype analog systems, developing digital expert/laser-optic systems, and linking developed microcomputer systems to a target mainframe. Phase 3 will concentrate on advanced technology associated with data transmission, high-definition television, and related integrating technologies.

Phase 1 began with a collaborative arrangement between UCF faculty and students and

the Lockheed Space Operation Company (LSOC). LSOC, under contract with NASA, has the overall responsibility for all prelaunch and postlanding processing of the space shuttle. Under the agreement, LSOC provides documentation of requirements and procedures of the current photographic close-out system, access to experts who have specific duties associated with space shuttle imagery, and entry into limited-access facilities when required for observation of current processing tasks and methods.

Several methodologies have been and will be used. Case study observations and interviews were used during initial systems analysis activities, followed by individual and group surveys designed to identify critical human and preperformance factors. Both structured interviews and written surveys were used. Similar measures will be used to determine the effectiveness of the prototypes when compared with traditional image processing methods.

Two-stage prototyping is being employed. First, a low-cost prototype will be constructed to collect data not directly related to the quality of the retrieved image. Information from this early prototype will then be used to construct a high-definition prototype that will validate all previous data and collect data specific to the quality of the retrieved image.

Phase 1: An expert photo retrieval system

To date, the research team has scrutinized the close-out process (i.e., authorization, collection, classification, and storage); space shuttle engineers who use the photographs have been interviewed and their comments recorded; and project start-up and purchasing actions have been completed.

On the technical side, a Pioneer Video laser disk player and Texas Instruments expert systems shell have been interfaced through a device-driver software program written in C. The driver controls the activities of the laser device. The working model demonstrates intelligent retrieval of still images and can be used to access motion sequences, computer-generated graphics, text, and multiple audio (i.e., language) channels.

The expert systems shell chosen for this prototype, Texas Instruments's Personal Consultant

Project Research Areas and Phases

Systems analysis

Year 1 Define the current dynamics of the case study system.

Survey existing users of close-out photography.

Year 2 Collect prototype system-improvement information from users.

Year 3 Evaluate a prototype high-definition television (HDTV) expert system.

Image systems (capture, storage, classification, retrieval, transmission)

Year 1 Develop a sample analog video data base of images.

Year 2 Develop sample medium-resolution digital data base of images.

Year 3 Develop picture-within-a-picture capability. Develop a sample high-definition video data base of images.

Alphanumeric systems (traditional data base, text data base)

Year 1 Define current classification and retrieval systems.

Collect existing data keyed to a sample image data base.

Year 2 Link the alphanumeric data base to associated micro and mainframe systems.

Year 3 Develop hypermedia links to the alphanumeric data base.

Expert systems

Year 1 Identify attributes of current expert systems shells.

Develop drivers to link expert systems to the sample image data base.

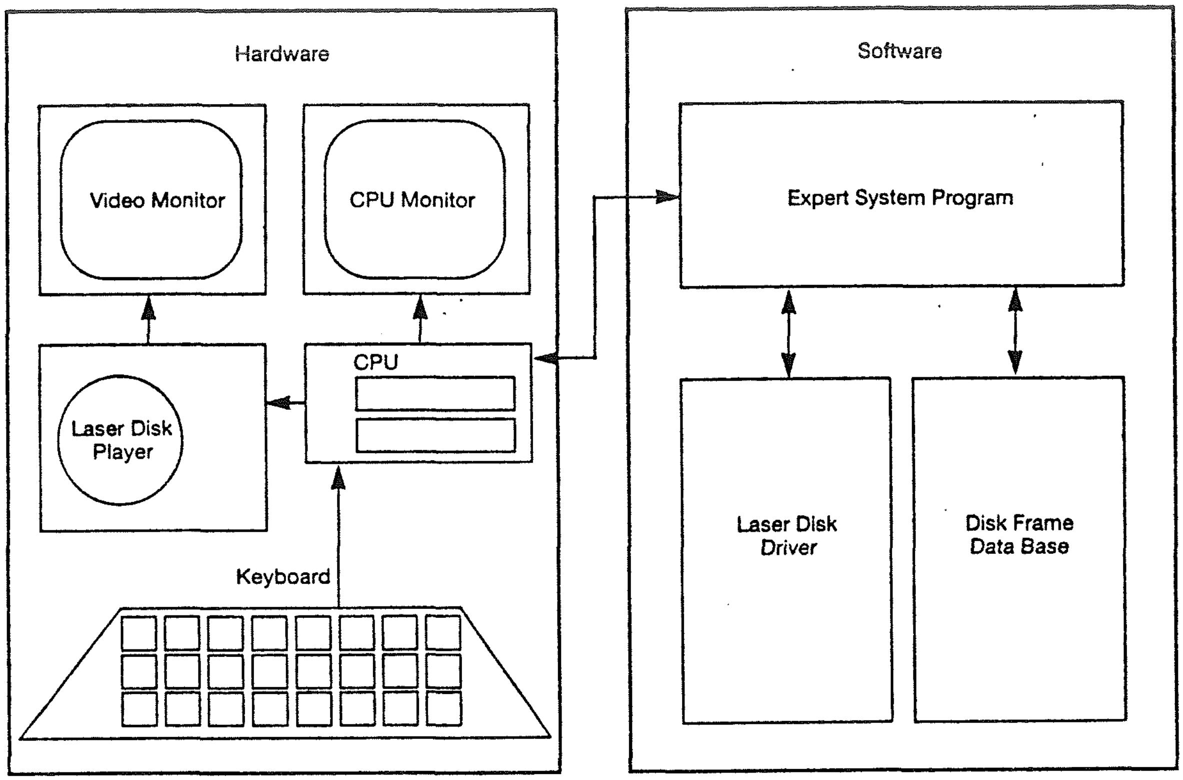
Develop a menu-driven expert system to control image definition data and classification entry.

Develop a menu-driven expert system to assist in retrieval of images.

Year 2 Develop natural language, query-based versions of phase 1 systems.

Year 3 Develop hypermedia links between expert systems and data bases.

Exhibit 1. *Prototype Expert Photograph Retrieval System*



Easy, has the capability to run external programs (.BAT, .COM, .EXE) directly from within the expert systems shell. Exhibit 1 shows the layout of this first analog demonstration system. A dBASE III+ data structure has been constructed that resides between the shell software and the laser disk's internal microcomputer. This data base allows information about the location of laser optical images to be entered directly and maintained through a standard DBMS by modifying expert systems rules.

MBA students from UCF's College of Business Administration have been working with operational personnel from NASA, the Kennedy Space Center, and contractor companies to construct prototype rule-based expert systems photograph classification and retrieval modules. These modules (which share a common dBASE III+ data base) implement an intelligent descriptor-based classification and retrieval system for sample close-out photographs. A full spectrum of commercially available expert systems shells (rule based and object oriented) are being investigated, as well as neural networks, data bases and query languages, and interface devices. Also under study are still video, hypertext, and Intel Corp's Digital Video Interactive (DVI) compact disk technology that can store large amounts of motion video.

Stay tuned

This research is expected to produce a series of standards that define an image DBMS for the collection, classification, storage, and retrieval of digitized color images of a quality users will find to be equal to standard 35-mm color slides. Issues related to analog versus digital collection, storage, and image display are yet to be resolved. Comparisons of expert systems and standard data bases to access images are still being weighed, as is the practicality of locating pictures within pictures

through image-based queries and other techniques. The case study prototypes that evolve from this applications research will, however, ensure that new technology can be integrated within current and planned NASA systems throughout the 1990s. ▲

Acknowledgments

Funding for this research would not have been possible without the vision and support of Tom Davis and Astrid Heard of NASA/Kennedy Space Center's Advanced Systems and Technology Office. The authors also wish to thank UCF co-investigators Ali Shaykhian of the College of Engineering and Ann Barron of the College of Education and the following College of Business Administration MBA students: Todd Shieffelin, Michael Vine, Russ Wielgos, Jeff Grossman, R.J. Johnson, Piage Leone, Brion Quandt, Stan Newberry, Dennis Duffy, Kent Golden, and Fred Damiano. Finally, Kevin Paule and Carolyn Beason of LSOC, Nancy Pope of NASA, and photo servicers Sherline Turbyville and Cathy King of TGS Technology Inc are thanked for their cooperation during the development process.

Notes

1. N.S. Chang and K.S. Fu, "An Integrated Image Analysis and Data Base Management System," *National Technical Information Service Report* TR-EE80-20 (May 1980).

Recommended Reading

Dayhoff, R.E. "A Medical Image Data Base System." *Computers in Healthcare* 9, 12 (December 1988).

Janus, S. "Hardware, Software Developers Striving for Better Image Data Base Management." *PC Week* 4, 8 (February 1987).

Meyer-Wegener, K.; Lum, V.Y.; and Wu, C.T. "Image Data Base Management in a Multimedia System." *National Technical Information Service Report* NPS52-88-024 (August 1988).

Appendix F

"Attacking the information access problem with expert systems."

Page intentionally left blank

Attacking the Information Access Problem with Expert Systems

James M. Ragusa and Gary W. Orwig

The growing information needs of many businesses and industries may soon exceed practical paper and magnetic storage capacities. Better storage and retrieval systems are clearly needed. Laser-optical devices (laser disks) under expert systems control can access masses of detailed visual information and standard text quickly, with the added element of interactivity. This article describes the joint applications research being done by the University of Central Florida and the National Aeronautics and Space Administration (NASA) that was introduced in the Fall 1990 issue of EXPERT SYSTEMS ("Expert Systems and Imaging: NASA's Start-Up Work in Intelligent Image Management").

Expert systems and laser-optical devices (laser disks) are being used independently in a wide variety of applications. In fact, each has evolved into a multimillion-dollar industry. When integrated, they can create greatly expanded interactive multimedia applications.

Efforts to interface laser disks and expert systems for intelligent interaction will allow large amounts of information (text and images) to be accessed almost immediately, thus alleviating the information access problem many businesses face. Application potential exists whenever there is a need for the rapid retrieval and low-cost archival storage of multimedia information. Integrated multimedia applications that have been somewhat successful include the commercial kiosk displays frequently seen in shopping malls and hotel lobbies. Kiosks contain the display equipment and the programs that promote interactivity. If designed properly, these public-access displays give users a sense that the system is responding to their needs.

More advanced interactive systems that can attack the information access problem are, however, the focus of both this article and a continuing NASA-sponsored applications research project. Twelve microcomputer-based expert systems shells and five laser-optical formats have been studied, and the general and specific methods of interfacing these technologies are being tested in prototype systems.

Expert systems shells

There are several ways that expert systems can be used within integrated software systems. Some expert systems can provide front-end control. In other cases, the expert system and other program elements share control through their integration. In the design of a prototype integrated system developed by the University of Central Florida and NASA, the expert system is the main controlling program. As such, its function is to call appropriate programs or subroutines that may be written in other languages. This design is possible because most expert systems shells provide facilities for calling other programs and accessing data bases or spreadsheets. For example, access capabilities enable users to extract information from a data base and pass it to the expert system's inference engine, which is a knowledge-

***James M. Ragusa** is an associate professor of management at the College of Business Administration, University of Central Florida, Orlando. Ragusa was formerly employed with NASA at the Kennedy Space Center. He can be reached at (407) 275-2915.*

***Gary W. Orwig** is an associate professor at the College of Education, University of Central Florida. He can be reached at (407) 281-5179.*

processing software subsystem.

Ten commercially available and relatively small microcomputer-based expert systems shells were selected for in-depth analysis. They are Personal Consultant Easy and Personal Consultant Plus (Texas Instruments Inc), Nexpert Object (Neuron Data), EXSYS (EXSYS Inc), 1st-CLASS (AICorp Inc), Guru (mdbs Inc), VP Expert (Paperback Software International), GoldWorks (Gold Hill Computers Inc), TIMM (General Research Corp), and Level5 (Information Builders Inc). By definition, these shells are expert systems without knowledge bases.

Two large hybrid tools—ART-IM (Inference Corp) and KEE-386 (IntelliCorp)—were later added to this list as microcomputer versions were made available. These popular and robust systems are widely recognized as significant development resources for the workstation environment, and each contains a full range of expert systems development capabilities well beyond those of the smaller shells. The features (retained in the microcomputer versions) include frames, semantic networks, object-oriented programming, blackboard architectures, and dynamic inheritance.

Shell features and interfacing capabilities

The main features of each of the rule-based shells are shown in Exhibit 1. Memory needs range from 256K to 10M bytes. The external interface feature of each shell is a crucial consideration when creating interactive multimedia applications. Although diverse, all the shells provide a mechanism that allows direct technology connection through ASCII codes so that data (e.g., the expert system's conclusions) can be passed to external programs (e.g., the laser disks).

Other parameters less crucial to this research effort are also shown in Exhibit 1, including each product's forward and backward chaining capabilities (i.e., the method of searching rules for drawing or supporting a conclusion), ability to access R:Base (Microrim) and vari-

Exhibit 1. *Expert Systems Shells and Their Features*

Shell	Minimum Memory (bytes)	External Interface	Forward Chaining	Backward Chaining	Access to R:Base	Access to dBASE	Access to Lotus 1-2-3	Average Retail Price
Personal Consultant Easy	640K	DOS-CALL command	×	×		×	×	$ 495
Personal Consultant Plus	640K	DOS-CALL command	×	×		×	×	2,950
Nexpert Object	640K	Execute command	×	×	×	×	×	5,000
EXSYS	512K	Run () command	×	×	×	×	×	395
1st-CLASS	256K	{ } command	×	×		×		495
Guru	640K	Run () command	×	×	×	×	×	6,000
VP Expert	384K	Call command		×		×	×	245
GoldWorks	6M	SYS:EXE command	×	×		×	×	7,500
TIMM	640K	Linkable objects	×	×				1,900
Level5	512K	Activate command	×	×		×		485
ART-IM	640K	System function	×	×				6,000
KEE-386	10M	LISP function	×	×				9,900

Exhibit 2. *Laser-Optical Systems and Their Characteristics*

Medium	Type	Market	Size	Capacity	Content
Video Laser Disk	Interactive	Training	8 in and 12 in	30 minutes	Analog video and sound
Compact Disk:					
CD-ROM	Interactive	Storage	5 in	650M bytes	Digital data, image, and sound
CDV	Noninteractive	Consumer	5 in	650M bytes	Analog video with 20 minutes of sound
CDI	Interactive	Consumer	5 in	650M bytes	Digital video, text, and sound
DVI	Interactive	Military	5 in	650M bytes	Compressed digital video
WORM Disk	Interactive	Military	12 in	18 minutes	Analog video and sound
Laser Card	Interactive	Business	Credit card	4M bytes	Digital text and graphics
Laser Film	Interactive	Military	12 in	18 minutes	Analog video and sound

ous dBASE versions (Ashton-Tate), and provisions for reading and writing to Lotus 1-2-3 files.

Although prices vary greatly, most expert systems shells have both development and delivery (run-time) versions available. An expert systems application can be created and tested using the development version, then distributed and used in an executable form. With these shells, the delivery version is always much less expensive than the development version.

Laser-optical devices

Five laser-optical formats were studied to determine their compatibility with the expert systems shells. Exhibit 2 summarizes information about these optical media, which are currently available or will soon be introduced on the market.

Current testing and evaluation indicates that there are no significant restrictions to the full integration of these devices with expert systems technology. All of the laser devices described can be linked to expert systems to create integrated applications.

Video laser disks. Video laser disks are produced in several forms; the most accepted is LaserVision, which reflects a set of specifications that laser disk manufacturers have established. An array of programs and players are commercially available in this format. The main drawback is that programs must first be produced on a master tape, which must then be sent to a special facility for conversion to disk. As a result, several noncompatible formats have been developed that allow direct recording from desktop recorders/players and are priced from $10,000 to $20,000.

Compact disks. Compact disks, the same format used for musical recordings, are used primarily as the storage medium for a variety of alphanumeric and image presentation systems. CD-ROM (compact disk–read only memory) is used for read-only storage of massive amounts of digital data. Information can be stored as images (up to 15,000), text (10,000 pages), or sound (30 minutes). CDV (compact disk video) is an evolving medium designed to present segments of motion video with high-quality sound.

Two new compact disk technologies are emerging. CDI (compact disk interactive) can present sound, data, still images, and motion. It uses a dedicated player that appears to be directed mainly at the consumer market. DVI (digital video interactive) is similar to CDI but is designed to work with advanced personal computers. It also appears to offer superior motion video at this stage of development. Currently, to produce any form of compact disk, finalized data tapes and videotapes must be sent to a specialized facility for conversion.

WORM disks. Write-once/read-many (WORM) disks, now available from several manufacturers, function as if they were very large capacity hard disk drives that can be written to and read from but not erased. Because of these characteristics, they are useful for storing archival data (e.g., business records) that need updating.

Laser cards. Laser cards are credit card–sized devices containing an optical stripe that stores digital information. A special reader connected to a monitor is used to access data.

Laser film. The primary advantage of laser film is that the disk, which is made of photographic film, can be recorded in real time and duplicated rapidly. The need for special equipment, however, has limited the medium's success.

Interfacing

Interfacing laser-optical devices and expert systems shells is accomplished with a device driver software written to control and query the functions of the laser device. The device driver program may be written in any language supported within the personal computer environment, including C, Pascal, Ada, FORTRAN, and LISP. To control the laser device from within the expert systems application, the expert systems shell must be able to access the laser device driver program. Typically, expert systems shells can call an external program that will perform an activity not supported by the shell. Each of the 12 expert systems tools listed in Exhibit 1 can run any other external programs (e.g., .BAT, .COM, or .EXE) directly from within the shell.

To control the laser device from within the expert systems application, the expert systems shell must be able to access the device driver software.

To provide full functional access to the laser device, the expert systems shell must be able to send standard commands as well as data through the laser driver. The commands tell the driver what action to request—for example, seek a still-frame display from a laser disk. To reduce delay times, the expert systems shell should allow data to be stacked in memory. If, instead, each command requires access to a hard disk or diskette drive, performance will suffer. With the exception of EXSYS, all of the expert systems shells studied can pass data to an external driver program through memory. EXSYS passes data to an external program by writing to a text file called "PASS.DAT". The called program—in this case, the laser driver—must read the "PASS.DAT" file to determine the requested action to perform.

Each expert systems shell formats data differently. Therefore, for device driver software to be shared by multiple expert systems shells, it must allow configuration selection for the shell used. Because most of the expert systems shells pass data in an ASCII format, the device driver should accept ASCII commands. In addition, it must convert the data to a correct form (configured for the shell selected) to process the commands. For example, to play a segment of a laser disk, the expert systems shell fires a rule that invokes an external program by passing the command syntax to play the disk. To perform this using Texas Instruments' Personal Consultant Plus shell, the following syntax is used: DOS-CALL "LASER" "PLAY FROM 1200 TO 1300". The external program "LASER" receives a request to play a segment of the laser disk from frames 1,200 to 1,300. To function properly, the laser disk–expert systems shell configuration must be selected as Personal Consultant Plus.

With expert systems shells that can access external data bases, such specification data as frame numbers can be stored in an external data base rather than in rules. This method is used in the latest version of the prototype integrated system.

The laser disk device driver interface

The laser-optical device selected for this research effort, an industrial-grade Pioneer LD-V8000 player, is controlled through a standard RS-232 data interface. Personal Consultant Plus is the expert systems shell in the current prototype. Laser disk operation is maintained by sending the laser disk player the proper ASCII character sequences. Exhibit 3 lists the control functions and their commands.

The laser disk control program is written in the C programming language for portability. The device driver accepts the commands that allow the expert systems shell to access and play still frames as well as specific segments of the laser disk. For this application, a commercially available laser disk was selected.

Exhibit 3. ***Laser Disk Functions and Commands***

Function Performed	Commands
Pause play action (picture ceases)	"PAUSE"
Start play action (audio and video signals played)	"PLAY FROM *** TO ***"
Step forward or backward a frame at a time	"STEP FORWARD" "STEP BACK"
Scan forward or backward	"SCAN FORWARD" "SCAN BACK"
Play at a specified speed	"SPEED ***"
Seek a particular frame address	"SEEK ***"
Display a still frame	"PLAY ***"
Control the audio and video signals	"AUDIO STEREO" "AUDIO CH1" "AUDIO CH2" "VIDEO ON" "VIDEO OFF"

Applications potential

Proven applications of expert systems technology can be found in business (including a large base of financial services), the aerospace and defense industries, education, the arts and sciences, and production and operations management. Several significant production-related expert systems have been put into operation in the US. These systems have one feature in common: they resulted in improvement of at least one order of magnitude over the original production method. Du Pont has an expert system for purging a distillation column of chemical impurities, and Westinghouse Electric has a system for diagnosing steam turbine generator problems. FMC Corp has introduced an expert system that helps prevent equipment failures and reduce waste production in the manufacturing of phosphorus. FMC's system also helps control costs. Digital Equipment Corp's VAX expert configurator, XCON, has spawned similar systems, including the truck configuration system used by Navistar International. Northrop Corp has a knowledge-based system for translating engineering drawings into a detailed plan for manufacturing complex aircraft parts.

Contemporary pressures and the need for expert systems applications are being addressed elsewhere in the world as well. In 1987, *Japan Computer Quarterly*, a publication of the government-sponsored Japan Information Processing Development Center, reported:

> **"In Japan we have developed what might be called 'expert systems fever.' In the world of Japanese industry, there are well over a hundred expert systems that are either already being put to practical use or soon . . . [will] be. Industries engaged most actively in the research and development of expert systems are steel, electric power, manufacturing (particularly the automobile industry), and construction. These are also the industries with the most expert systems in operation. Expert systems now under development are advanced enough for use in strategically important applications like design and planning."**

Within the Japanese market, steelmaker Nippon-Kokan has an expert system that interprets data from thousands of blast-furnace sensors. Toyota has developed a system to advise high-level technicians on automobile diagnosis and repair. Canon uses an expert system to design zoom lenses, and Kajima has developed an expert method for selecting piling material for building foundation construction.

Interest in integrated applications

In many respects, the applications research described in this article is an extension of a theme introduced in 1988 at the American Production and Inventory Control Society's international conference. Focusing resources for competitive advantage was the basic theme chosen by the conference committee. Today, interest is being shown in integrated knowledge-based, laser-optical systems that can help

Exhibit 4. ***Potential Integrated Applications***

Consumer Services
Service center advice
Car buying advice
Do-it-yourself home and car repair
banking service advice
Real estate selection
Plant care advice
Interior design advice

Travel, Tourism, and Recreation
Vacation selection
Outdoor recreation and camping site selection
Entertainment options
Hotel and restaurant selection
Map and route guidance
Golf course selection

Law Enforcement
Mug shot identification
Evidence inventory
Prison inmate education
Accident reconstruction
Vehicle identification
Crime scene records
Evidence presentation

Human Resources
Employee orientation
Loss prevention training
Company orientation
Skills training
Management development
Customer orientation

Marketing
Product identification and display
Customer and dealer information
Convention center selection
Electronic catalogs
Sales presentations
Distribution advice

Education and Training
Intelligent computer-aided instruction
Medical and paramedical training
Military simulation training
Customer service training
Teacher preparation
Adult literacy education
Language training
Library training
Sex education

Other
Architectural design assistance
Accounting audit reference storage and retrieval
Public works design
Urban planning
Landscaping

companies achieve a competitive advantage (see Exhibit 4 for potential integrated applications.

NASA and its contractors recognize the promise and advantages of linking these technologies as they develop applications for current and future space shuttle programs. NASA and the Kennedy Space Center have provided research applications funding to the University of Central Florida to explore the potential for the intelligent classification, storage, and retrieval of shuttle ground-processing close-out photography. NASA's need is significant because approximately 100,000 photographs have been obtained for each of the 35 shuttles processed at the Kennedy Space Center to date. Other NASA applications yet to be explored include intelligent computer-aided instruction, simulation and training, human resources development, and performance measurement. Additional processing-related opportunities include inventory management, planning and control, operations management, purchasing, quality control, distribution, and logistics.

A visual industrial data base

Video laser disks have been synonymous with interactive applications for many years now. Several years ago, Digital Video Corp worked with Westinghouse's transportation division on a prototype interactive parts maintenance system. This hardware and software system used standard alphanumeric data bases and a visual data base on video laser disk to allow engineers to quickly identify and order replacement parts for light-rail transit systems. The visual data base consisted of a pyramid structure of enhanced technical drawings. AT the top of the pyramid was a top-level or overall drawing of a transit car. Successive layers of the pyramid structure provided additional levels of detail for the functional units of the cars, with the final or bottom layer consisting of images of individual nuts, bolts, and indicator lamps. Using a light pen, an engineer could point to areas of the screen drawing where detail was needed, and a lower-level image would appear for that part of the transit car. After a few selections, the engineer could have a picture of the needed module or device; an adjacent monitor provides all the necessary ordering information from the standard alphanumeric data base.

Although field testing of this experimental system had highly positive results, the system had its limitations. Engineers often wanted to

take alternate paths through the images, or they wanted to view collections of images that contained similar components but in different areas of the transit car. When this system was developed, microcomputer-based expert systems shells were not readily available. If the system were developed today, alternate search strategies would be much easier to prototype and implement.

Production and operations

In production and operations, the applications potential is broad. A partial list of applications includes product planning, manufacturing resource management, and process design (e.g., computer-integrated manufacturing, location and layout selection, and materials and inventory management). Master production scheduling (MPS), material requirements planning (MRP), just-in-time (JIT) inventory, project scheduling and control, shop floor control, quality management and control, and distribution and logistics applications are also possible.

The area of process design, for example, includes the selection of input, operations, work flows, and methods. Assuming an expert system based on the knowledge of the organization's expert process designer were available, another designer (not necessarily the best design expert) could begin the design process. Connected to a laser-optical device, complete access to motion and still sequences, text and graphics, and sound would be possible. The current process, potential problem areas, available resources, vendors, work flows, and existing methods could be reviewed. What-if interaction with the resource base of knowledge and information would be a significant asset for this process. The results of an analysis could be recorded for future use to a WORM disk or laser-card device.

In another situation, integrated technology could be developed for the high-level training of new personnel or the upgrading, recertification, and cross-training of existing personnel. If the training applications were equipment setup or changeover problem solving, motion and still sequences, text and graphics, or sound could be employed in the program. Interactive computer-based training would allow standardized instruction, correction, testing, and remediation. If equipment setup is the target, previous methods and techniques could be displayed and described by the system. After testing and evaluation, the trainee's performance could be determined by the expert system's evaluation algorithm with scores recorded on the laser data base. The end result would be standardized training, flexibility, and improved performance with reduced participation by human instructors.

Conclusion

This article presents results of applications research directed at finding an improved method of storing and accessing information. One solution involves the synergistic interfacing of expert systems shells and laser-optical devices. A prototype system has been developed to prove the feasibility of integrated systems. They are not only possible but desirable if a competitive advantage is to be achieved. ▲

Acknowledgments

This technology transfer–based research is a result of grants from the University of Central Florida's Institute for Simulation and Training (IST) and NASA Kennedy Space Center's Advanced Projects, Technology, and Commercialization Office. Special thanks go to Michael Companion, the IST project monitor, and to NASA Kennedy Space Center's Astrid Heard. Others involved in this project are Ann Barron (College of Education) and Ali Shaykhian (College of Engineering). The following research associates must also be thanked: Michael Vine, Todd Schieffelin, and Russ Wielgos (all from the College of Business) and Hoi Yoo (College of Engineering).

Recommended Reading

Feigenbaum, E.; McCorduck, P.; and Nii, H.P. *The Rise of the Expert Company: How Visionary Companies Are Using Artificial Intelligence to Achieve Higher Productivity and Profits.* New York: Time Books, 1988.

Krjewski, L.J., and Ritzman, L.P. *Operations Management: Strategy and Analysis.* Reading MA: Addison-Wesley, 1987.

Ragusa, J.M., and Orwig, G.W. "Expert Systems and Imaging: NASA's Start-Up Work in Intelligent Image Management." *Expert Systems: Planning/Implementation/Integration* 2 (Fall 1990).

Appendix G

"Object oriented databases"

OBJECT-ORIENTED DATABASES

by

Russell A. Wielgos

November 8, 1990

TABLE OF CONTENTS | PAGE

INTRODUCTION

Traditional database systems were developed in the mainframe environments of the early 1960s. They were usually applied to numeric and record-based data, such as employee records and parts inventory. During the 1970s, on-line information systems became popular and were used for full-text searching of textual databases. Expert systems technology was developed in the mid-1970s and then, in the late 1970s and early 1980s, techniques related to object-oriented programming grew out of advances in software engineering, user interfaces and high-level languages.[1]

These object-oriented concepts offer significant advantages to the programming and data modeling business. Some of these advantages include the ability for user programming, easier maintenance and updating of software, and the ability to share data and procedures among applications and networks in a modular, reusable fashion.[13]

The basic idea behind the object-oriented approach is that objects are seen as self-defining components that incorporate not only data, but also, descriptions of their behavior and of their relations with other objects. Object classes can be divided into subclasses via inheritance. The concept of inheritance, which will be described more fully later on, allows similar procedures to be invoked without all new code needing to be written for each.

Also, because objects can be manipulated and altered without having to rewrite any code, the benefit of true end-user programming is built into object-orientation.[13]

Another unique characteristic of the object approach is that it more closely simulates the way people think and function than do procedural systems. It is a method of telling the computer how to conceptually represent and manage information about the real world.[3] This helps to free individuals from the need to think like a computer, instead, encouraging groupings according to common-sense categories and procedures.[13]

Foundational object-oriented concepts have evolved in three different disciplines: first in programming languages, then in artificial intelligence, and then in databases.[8] This paper will focus on the database concepts of object-orientation, while also, briefly touching on the artificial intelligence concepts.

OBJECT-ORIENTED DATABASE CONCEPTS - NATURE OF THE TECHNOLOGY

In object-oriented databases, an object is a thing which exists and has identity, such as a microwave located in the kitchen or Richard Nixon. These objects, unlike an item in a relational database, are persistent, i.e., they can survive multiple sessions or transactions.[12] Every object has a specific state and behavior. The state of the object is defined as the set of values for the attributes (i.e. ON/OFF, etc.) and the behavior of an

object is the set of methods, or programming code, which operates on the state of the object.[8]

A group of related objects make up an object class.[14] A microwave oven, a group of rooms, or a set of people, are all examples of object classes. Objects belonging to the same class are called instances of that class. A class describes the form (attributes) of the instances, and the operations (methods) applicable to each instance.

Some of the other fundamentally important aspects of object-oriented databases are the concepts of inheritance, object identifiers, messages, encapsulation and version management. These concepts will be discussed next.

All subclasses of a class inherit all of the properties defined by that class, and can also have additional properties local to them.[17] This powerful concept, called inheritance, is a technique for obtaining information from an object's parent class. It is helpful in that it reduces the need to specify redundant information, while also, simplifying the updating process, as information about many object instances can be changed and entered on a single update action.[5]

In the concept of object identifiers, object-oriented databases are very much similar to network or hierarchical databases. They each use the notion of pointers.[12] The object identifier is a logical pointer which is used to pinpoint an

object to retrieve.[8] It is the property of an object that distinguishes it from all other objects and is independent of content, type and addressability.[9]

A message represents the interaction between objects.[5] Instead of naming a procedure to perform an operation on an object, the object is identified and a message is sent to it. A selector in the message specifies the operation to be performed and the object that receives the message is then responsible for deciding how to respond using its own methods for performing the requested operation.[10]

Encapsulation is an abstraction mechanism which accesses the object and modifies it only through external interface routines and operations. The internal implementation details, data structures, and storage elements used to process the object and its operations are invisible to the user, thus helping the object-orientation to reflect reality more closely.[9]

Another important concept of object-oriented databases is the concept of version management. In some applications, the same object undergoes multiple changes or state transitions, and it is often desirable to access or investigate previous states of the object.[7] This version management is handled by archiving old versions of the objects and later returning them, using their unique identifier and time-stamp or version number.[5]

OBJECT-ORIENTED VS. RELATIONAL DATABASES

Currently, less than 10 percent of the World's computerized data is stored in a data base.[5] This fact, most likely, is due to the inability of currently available database management systems (DBMS) to provide the data modeling and retrieval facilities required by most computer users.

The "Third Generation Database System Manifesto", authored by several relational database experts, decrees that the 1970s was the era of the first generation - the hierarchical database; the 1980s was the era of the second generation - the relational database; and that the 1990s will be the era of the third generation - an extension of the relational database.[12]

According to the Manifesto, the third generation database will have 3 basic requirements:

1.) It must accommodate a broader range of data types, such as images, multimedia documents, video and other "objects".
2.) It must support the positive features found in the second generation relational databases, such as non-procedural access and data independence.
3.) It must communicate and be interoperable with distributed DBMSs, C programs, business applications, UNIX commands, software engineering tools, etc.[12]

These relational database experts never even mentioned the words "object-oriented", thus the battle has begun. On one side the evolution seekers, i.e. the relational database guys vs. the revolution seekers, i.e. the object-oriented guys.

Vendors of object-oriented database systems such as Object Design Inc., Ontologic Corp. and Servio Corp. claim that their products perform more than twenty times better than relational databases for such applications as CAD and CASE.[20] They would like to see their technology replace all the less effective relational databases.

The relational database supporters feel, that if a change is necessary, that the object-oriented technology should be added on to their time-tested relational systems, not replace them. According to David Beech, senior architect at Oracle Corp, one of the first relational database suppliers, "The object model is just an extension of the relational model, and having yet another type of database system in shops raises a lot of problems."[20]

Object-oriented vendors refute this by replying, "We've built a Porsche from the ground up, while they're still strapping boosters on a Ford Fairlane and saying it will go just as fast."[20]

Perhaps the best way to solve this type of confusion is to look at the various differences in the two database models. Lets give it a try.

Besides the fact that object-oriented databases introduce a whole new set of concepts, such as the aforementioned inheritance, class, method and message, there are some other very real and clear distinctions between the two databases.

For example, relational databases are based on a mathematical

theory, while object-oriented databases use a navigational model of computation. The relational crowd criticizes this navigational model by asking, "How do they traverse complex structures (objects)?" They claim that it is a "back-to-the-future" technique from the world of hierarchical databases. Object-orienters argue that the advantage of navigation is that it is easier to weave your way through objects that model the real world rather than tables, tuples and records, which do not allow for a nested structure and view of a design.[12] Furthermore, there are important applications from computer-aided design and artificial intelligence which absolutely MUST navigate through large databases.[8]

Another difference between the two is that relational databases are too limited for a large class of applications. These applications, characterized as complex, large scale, and data intensive, need a database model that is more expressive and flexible than a relational model. Object-oriented databases are being developed to meet the data handling needs of such applications. This is helped by the way object-oriented databases can store class definitions and instantiations of these classes.[6]

The way applications must query information in a database is also handled quite differently between the two models. In the relational model, an application sends a query to a database which then returns a number of values. These values are typically stored in the application's data structures, manipulated in these structures, and then shipped back to the database to be stored.

This flat nature of relations makes it difficult to model complex object networks. Instead, the mimicking of hierarchies with identification keys instead of pointers or direct references are necessary.[6]

In the object-oriented world, however, a program sends a message to an object in the database to manipulate its data, compute some value, and/or to return something, such as a reference to another object in the database.[6]

In a related issue, because complex objects are stored in a series of relations, a number of queries need to be applied to retrieve a single object from a relational database. These queries are applied sequentially, with one query depending on the result of the previous query. On the other hand, with an object-oriented database, sending one message can take the place of performing many relational database queries. In other words, messages can invoke other messages, thereby alleviating the problem of sequential queries in relational databases.[6]

In terms of relational schema, relational database systems represent the database schema in the form of a set of relations, including a relation for all other relations in the database. This schema is much more difficult to represent and maintain in an object-oriented database, since the schema is no longer a simple collection of largely independent relations, but a collection of classes which are interrelated to one another through generalization and aggregation relationships.[8]

Hypermedia, being a hot topic of the day, is also being brought up as a definite fault of relational databases. Unlike object-oriented databases, relational databases cannot provide direct, natural representations of graph-structured object spaces. Because object spaces are flat tables, complex relationships between multimedia products and objects cannot be expressed directly.[9] This may serve to be a major flaw of relational databases in the near future.

A number of other differences between the two databases models exist, surely enough to concentrate an entire paper on, but here only some of the more obvious ones were brought out. From this discussion, the reader can see that each model has its own set of strengths and weaknesses, and each model should only be evaluated based on the application to serve.

USERS OF OBJECT-ORIENTED DATABASE TECHNOLOGY

People who are developing object-oriented applications will increasingly want to store and retrieve instances in databases.[4] The main target, thus far, of object-oriented databases has been the engineering, computer-aided design (CAD), computer-aided manufacturing (CAM), computer-aided software engineering (CASE), and office automation markets. These applications all have characteristics different from traditional business or accounting applications. CAD, CAM, CASE and office

automation applications use such large amounts of data that it is nearly impossible to cluster the application's related data.[7] A number of these object-oriented database applications and prototypes will now be described in more detail.

One of the earliest object-oriented database products was G-BASE which was introduced by Graphael in 1987. It is a Lisp-based system with entities and methods defined using Lisp syntax. G-Base supports abstract data typing and multiple inheritance of classes. The G-Base environment also includes an interactive graphical object browser, interactive report generator and declarative query language. G-Base, also, provides elegant modeling, browsing, and querying capabilities for bulk data, but the early versions did not support multi-user transactions or versioning.

The Gemstone object-oriented database, from Serviologic, was one of the earliest to provide strong support of object identity, inheritance, and encapsulation, coupled with many database capabilities such as persistence, transactions and ad hoc querying.[7] Gemstone was created with components of C and Smalltalk.[3]

Iris, developed at Hewlett-Packard, is a next generation DBMS research prototype which supports inheritance, abstract data and referential sharing of objects.[19] It is based on functional data models and includes an object-oriented extension

of SQL, called Object SQL.[7] Iris, designed to meet the needs of rich modeling constructs, also supports direct data support for inference, novel data types, lengthy dynamic data base interactions spanning over days and many versions, and interfaces with many object-oriented programming languages.[2] It was built with the intent to meet the needs of new and emerging database applications such as office automation and knowledge engineering, engineering test and measurement, and hardware and software design.[22] Each of these applications require a number of capabilities which are not supported by the current generation relational databases.[19]

OZ+, developed at the University of Toronto, merges object-oriented programming concepts derived from Smalltalk and is intended for use in modeling office activities which will provide efficient methods for storage and retrieval of persistent data in a multi-user office. OZ+ employs object-oriented programming concepts with a slight variation, such as objects, contents (the object's data), and rules (the object's operations), events and messages. The OZ+ system prototype was implemented on a SUN 3/50.[2]

And finally, the ORION prototype object-oriented database system, developed at Microelectronics and Computer Technology Corporation, is a single-user, multitask database system intended for applications in artificial intelligence, multimedia documents for office automation, and computer-aided design. It is filled

with advanced features, such as version and change notifications, transactions management, and multimedia management. Orion was implemented in Common Lisp on a Symbolics 3600 machine and on a SUN UNIX workstation.

As expected, with object-oriented databases being relatively new to the business world, there is not yet a large following or a great number of users for the object-oriented database. The prototypes, discussed above, have served as great research in the area and will eventually prove to be quite helpful in publicizing this new technology.

MIS OPPORTUNITIES FOR OBJECT-ORIENTED DATABASES

As previously stated, object-oriented databases are currently being used for such applications such as computer-aided design (CAD), computer-aided software engineering (CASE) and office automation. These types of applications may also, occasionally, fall under the category of MIS applications.

Object-oriented databases attempt to satisfy the needs of users whose applications need the following features:[5]

- a variety of data types and type constructors
- modelling accuracy
- derived data
- set value attributes
- the ability to model actions

Object-oriented databases do not attempt to solve acknowledged data-modeling problems such as:[5]

- temporal and spatial logic
- incomplete information
- conflicting information
- system self-knowledge
- common-sense knowledge

Much of the research and use of object-oriented databases, done until now, has been related to more technical areas than MIS and business. Due to the complexity of the technology, I do not see this changing very much, in the near future, except in cases of applications needing the above features. I believe that relational databases or their third generation "big brother" will fulfill most of the World's MIS needs for years to come.

ECONOMIC IMPACT OF OBJECT-ORIENTED DATABASES

The one big economic impact, which I view object-oriented databases to have on the computing industry, is the decreasing reliance on the computer programmer. This trend, helped along by the arrival of the fourth-generation languages, will continue with object-oriented databases, due to the technology's ability to put the application programming in the hands of the user. This will leave the programmer more expendable, at the application level.

Object-oriented databases will also be able to save corporations money, due to its improvements over the current relational database technology. Any technology which is more

efficient and serves the company's purpose better than before, will end up saving the company money, in the long run.

FUTURE OF OBJECT-ORIENTED DATABASES

Object-oriented databases are still an excellent area for further research and standardization. First of all, there is a clear need to formalize and/or at least standardize the object-oriented concepts, if a true foundation is to be laid.[8] There also exists no formal model for queries made to object-oriented databases.[17] In order for this technology to be widely accepted, these fuzzy areas must be rectified and according to researchers, various standardization attempts are currently underway.

Object-oriented databases are also moving into the area of hypermedia. In the proposed Object Composition Petri Net (OCPN) model, researchers are attempting to approach the area of object storage and retrieval for multimedia synchronization. This model is expected to reveal an elegant approach to interfacing general multimedia services to databases.[18]

Another exciting area which object-oriented databases may move into, in the near future, is the area of intelligent databases. An intelligent database is one that "manages information in a natural way, making that information easy to store, access and use".[9] The emphasis is on information rather than data because these databases incorporate not only traditional applications, such

as inventory management, but also automatic discovery systems, imaging applications and knowledge bases.[9]

According to Khoshafian, "In the evolution of databases and data models, object-oriented databases form an important and necessary phase. The future of databases, however, is in intelligent databases."[7] Intelligent databases represent a new technology for information management that has evolved as a result of the integration of such diverse fields as: object-oriented concepts, expert systems, hypermedia and information retrieval.[7] (See figure 1). Khoshafian predicts them to be the single most important component in the integrated environment of the 1990s.[9]

EXPERT SYSTEMS

OBJECT ORIENTATION

INTELLIGENT TOOLS

INTELLIGENT DATABASES

INFORMATION RETRIEVAL

HYPERMEDIA

TRADITIONAL DATABASES

FIGURE 1

CONCLUSIONS

As this paper shows there are still many hours of research to be spent before object-oriented databases will be the savior that some may see it to be. There is a strong need for object-oriented database standards to be drawn up so that all research can end up serving the same coordinated purpose. Perhaps this would help speed up the research process.

Having served as a computer professional myself, brought up in the era of structured programming and assembler, I find object-oriented concepts to be a whole new and sometimes confusing technology, but it certainly may have some advantages over relational types. This will be left for the researchers to decide, but PC Week did have this prediction:

> "Few observers expect object-oriented database management systems to replace relational databases. ODBMS outperforms relational systems for handling complex data such as engineering diagrams and bitmapped graphics, but relational databases are better for transaction-oriented applications such as account updates," said Rick Cattell, a distinguished engineer and ODBMS expert, at Sun Microsystems Inc. "I think the technology is going to take off in a limited domain. The market that these folks are initially addressing is maybe ten percent the size of the business data-processing market."[20]

As with many other recent technologies, such as artificial intelligence or neural networks, it again looks as though only time will tell, if and when, this object-oriented technology will end up turning into the helpful technology we hope it will, or end up being lost in the shuffle, as was the case for many of the early robotics efforts. Hopefully, unlike robotics, object-oriented databases will live up to its potential.

ANNOTATED BIBLIOGRAPHY

[1] **J. Shandle, "Will Vendors Line Up Behind Objectivity?", Electronics, April 1990, pp. 66-67**

This article explains the vendors' point of view on the object-oriented data base technology. Basically, it says that the technology is still two years from being widely incorporated in the product sectors and it will definitely be helpful in the CAD and CASE applications.

[2] **David Hu, "Object-Oriented Environment in C++", Management Information Source, Inc., 1990, pp. 456-466**

This book discusses the object-oriented environment in the C++ programming language. The pages referenced spoke of object-oriented DBMSs and intelligent databases. Several object-oriented database prototypes are discussed.

[3] **J. McCaskey, "Object-oriented database keeps the house in order", Electronic Design, March 19, 1987, pp. 129-134**

This article discusses the strengths of object-oriented databases and programming and explains their use in the simulation world.

[4] **P. Harmon, "Object-Oriented Systems", Intelligent Software Strategies, Volume VI, No.9, pp. 1-20**

This monthly newsletter on expert systems, OOP, CASE, Neural Networks and Natural language, put out by the Cutter Information Corp., covers such topics as object-oriented: systems, programming, products, CASE tools, interface tools, CAD/CAM development tools and product vendors.

[5] **M.A. Garvey and M.S. Jackson, "Introduction to object-oriented databases", Information & Software Technology, December 1989, pp. 521-527**

This paper is a tutorial introduction to object-oriented databases. It describes the background of object-oriented databases and outlines the rationale for this approach. Applications are also discussed.

[6] **K. Smith, and S. Zdonik, "Intermedia: A Case Study of the Differences Between Relational and Object-Oriented Database Systems", OOPSLA '87 Proceedings, Oct. 4-8, 1987, pp. 452-465**

This paper compares two approaches to meeting the data handling requirements of Intermedia, a hypermedia system. Intermedia, written using an object-oriented programming

ANNOTATED BIBLIOGRAPHY, CONTINUED

[6] cont language, relies on a traditional relational DBMS for data storage and retrieval. The ramifications of replacing the relational database with an object-oriented database are also discussed.

[7] **S. Khoshafian, "Insight into Object-Oriented Databases", Information & Software Technology, May 1990, pp. 274-289**
This article explains that object-orientation provides a more direct and natural representation of real-world problems. It explains the basic concepts of the technology and also discusses intelligent databases, in depth.

[8] **W. Kim, "Object-Oriented Databases: Definition and Research Directions", IEEE Transactions On Knowledge Engineering, Sept. 1990, pp. 327-341**
This article gives a basic foundation of the object-oriented database using results obtained from the ORION series of object-oriented database systems. Future research direction is also discussed.

[9] **S. Khoshafian, K. Parsaye and H. Wong, "Intelligent Database Engines", Database Programming and Design, July 1990, pp.56-65**
This article explains how DBMS technology will evolve in order to handle such new technologies as hypermedia, object orientation and other new business needs. Relational, object-oriented and intelligent databases are discussed. The factors associated with object-oriented databases are also explained.

[10] **F. Manola, "Object-Oriented Knowledge Bases, Part I", AI Expert, March 1990, pp. 26-36**
This article explains the history of DBMS concepts and explains how object-oriented concepts can be used to address the requirements of KBIIS systems and information processing resources.

[11] **F. Manola, "Object-Oriented Knowledge Bases, Part II", AI Expert, April 1990, pp. 46-57**
This continuation article explains the DOM (distributed object manager) technology and its tools and applications. Knowledge-based processing and HDDBMS intergration are also discussed. Object-oriented database technology's application in integrating KBIIS, is also on the agenda.

ANNOTATED BIBLIOGRAPHY, CONTINUED

[12] C. Stone and D. Hentchel, "Database Wars Revisited", Byte, October 1990, pp.233-242

The article explains the differences between relational and object-oriented databases and describes how experts feel about the eventual third generation databases.

[13] J. Blackford, "The Story Of 'O'", Personal Computing, June 29, 1990, pp. 83-85

This article gives a brief overview of the object-oriented concepts, including databases, programming languages, operating systems and environments.

[14] E. Sibley, "Relational Database Design Using An Object-Oriented Methodology". Communications of the ACM, April 1988, pp. 414-424

This article explains the object modeling technique approach to relational database design. A comprehensive explanation and two applications show the semantic improvements of OMT over other approaches. Basic object-oriented approaches are also discussed.

[15] K. Lindsay, "Expert Systems in the CIM Environment", Sterling Publications, 1987, pp. 6-7

This Intellicorp Technical Article provides a comprehensive view of Computer Aided Manufacturing (CIM) and discusses some of the techniques, developed within the field of AI, which can be applied to accomplish the computer integration of a manufacturing enterprise. AI, object-oriented data management and CAM Net are also discussed.

[16] F. Staes and D. Vermeir, "Browsing a la carte in Object-Oriented Databases", The Computer Journal, Vol.32, No.4, 1989, pp. 333-340

This paper presents a user interface to the KIWI object-oriented database system. OOPS+, browsing and querying are also discussed.

[17] J. Banerjee, W. Kim and K. Kim, "Queries In Object-Oriented Databases", IEEE, 1988, pp. 31-37

This article develops a model of a query under an object oriented data model and analyzes the fundamental differences between queries in relational and object-oriented databases.

ANNOTATED BIBLIOGRAPHY, CONTINUED

[18] **T. Little and A. Ghafoor, "Multimedia Object Models for Synchronization and Databases", IEEE, 1990, pp. 20-27**
This article proposes a technique for formally specifying and modeling the temporal composition of multimedia data. The proposed model is based on the Timed Petri Nets and the logic of temporal intervals.

[19] **W. Kim and F. Lochovsky, "Object-Oriented Concepts, Databases and Applications", ACM Press, 1989, pp. 219-340**
This book discusses all aspects of the object-oriented technology, including databases, programming, modeling and applications.

[20] **P. Sherer, "ODBMS Upstarts Challenge Relational Database Ground", PC Week, Oct. 15, 1990, pp. 61,65**
This article discusses the differences between the relational database followers and the object-oriented followers. The article gives both sides of the quarrel of which type of database will end up being the next third generation database.

[21] **H. Afsarmanesh and D. Mcleod, "The 3DIS: An Extensible Object-Oriented Information Management Environment", ACM Transactions On Information Systems, Oct. 1989, pp. 339-377**
This article explains the 3-Dimensional Information Space (3DIS) object-oriented framework for information management and how it is specifically oriented toward supporting the database requirements for data intensive information system applications. The article describes the related research and object specifications, as well.

[22] **D. Fishman, D. Beech and H. Cate, "IRIS: An Object-Oriented Database Management System", ACM Transactions On Information Systems, Jan. 1987, pp. 48-69**
The IRIS DBMS is a research prototype of a next generation DBMS intended to meet the needs of new and emerging database applications, including office automation and software design. This article discusses IRIS in detail, including its content, needs and capabilities.

[23] **M. Hernick and S. Zdonik, "A Shared, Segmented Memory System for an Object-Oriented Database", ACM Transactions On Information Systems, Oct. 1989, pp. 70-95**
This article describes the basis of a object-oriented database and the basic architecture of the system implementing it. Segmented and transaction schema of architecture are present, as well as, object access, replication and clustering.

Appendix H

Potential applications: Interfaced expert systems/laser optics technology

POTENTIAL APPLICATIONS
INTERFACED EXPERT SYSTEMS/LASER OPTICS TECHNOLOGY
As of: January 25, 1991

APPLICATION # 1: NASA Shuttle Close Out Photography

ORGANIZATIONS: LSOC, USBI, Martin Marietta

NASA CONTACT: Astrid Heard, PT-AST, 867-2780; Nancy Pope, TE-CID, 867-2090

USER CONTACT: John Seaman, LSO-157, 861-2396; Kevin Paule, LSO-201, 867-0726; Gene Cain, Analex, 867-5716

NEEDS: To provide for the storage and rapid retrieval of Shuttle close-out photography. Approximately 10,000 8" x 10" color photographs obtained per launch.

MEDIA: 8" x 10" color still photographs.

SOLUTION: Use an expert system front-end linked to laser optics devices for photo information classification and retrieval. Baseline still video technology.

STATUS: Under active cooperative agreement development.

* *

APPLICATION # 2: Press Site News Photo Support

ORGANIZATION: NASA-KSC Public Affairs

USER CONTACT: Bill Johnson, PA-PIB, 867-7819/2468; JoAnn Mattey, TGS, 867-7826

NEEDS: To provide rapid retrieval of KSC-related photographs for various world-wide news organizations. Approximately 20,000, 8" x 10" color and B & W photographs; and 3000, 4" x 5" positive color transparencies exist in the system.

MEDIA: 8" x 10" color and B&W still photographs; and 4" x 5" positive color transparencies.

SOLUTION: Use an expert system front-end linked to a analog laser optics player for photo information classification and retrieval.

STATUS: This application will be used for validation of

expert system/laser optical storage, search and retrieval of a large database of photographs. Four (4) College of Business MBA students have developed a photo classifier and photo retriever during the Fall 1990 semester. In addition, one (1) College of Engineering student is developing a related audio/visual tape library database during the Fall 1990 and Spring 1991 semesters.

* *

APPLICATION # 3: LSOC Engineering Data Access & Management System (EDAMS) Retrieval Systems

ORGANIZATION: LSOC

NASA CONTACT: TE

USER CONTACT: Hudson Haile, LSO-043, 861-2261

NEEDS: To reduce production time and risk associated with processing, launch, and recovering Space Shuttle vehicles. Timely, adequate, and accurate engineering data are frequently needed. The desire is to create an integrated, open architecture system using the existing installed base of Shuttle processing engineering data.

MEDIA: Drawings, schematics, test data, specifications, photos, tables, historical data, etc.

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: Databases for several segments under discussion. Digital technologies will improve applications potential.

* *

APPLICATION # 4: KSC Lightning Data Storage and Retrieval

ORGANIZATION: NASA-KSC

USER CONTACT: Bill Jafferis, DF-ESS-22, 867-4438

NEEDS: To store and retrieve lightning satellite, radar, and field mill data (approx. TBD images). Approx. 40,000 satellite images and 10,000 radar images exist. Very large quantities of field mill networked data from 34 KSC sites stored on magnetic tape. Data is not readily available for analysis.

MEDIA: Satellite and radar photographic and images 8" x 11" strip chart hard copy images.

SOLUTION: Use an expert system front-end linked to an analog laser optics player for image retrieval.

STATUS: Satellite, radar, and field mill data has been recorded on laser disc system. Resolution is a minor problem but retrieval using an expert system/database is feasible for up to 54,000 images.

* *

APPLICATION # 5: NASA Educator Resource Center Slide Retriever

ORGANIZATION: NASA-KSC

NASA CONTACT: Steve Dutczak, PA-EAB, 867-4444

USER CONTACT: Janet O'Sullivan, Omniplan, 867-4090

NEEDS: To provide for a retrieval system for 35 mm color slides. Slides are used by national and international teachers in their classrooms.

MEDIA: 35 mm color slides (approx. 550).

SOLUTION: Use an expert system front-end linked to a analog laser optics player for photo information retrieval.

STATUS: Several database systems are being developed by two (2) College of Engineering students during the Fall 1990 and Spring 1991 semesters. They include systems for teacher and resource documentation, and inventory and mail-out.

* *

APPLICATION # 6: Univ. of Florida Citrus Tree Census Project

ORGANIZATION: NASA-KSC/Univ. of Florida Institute of Food & Agricultural Sciences.

NASA CONTACT: Tom Hammond, PT-TPO-A, 867-3017

USER CONTACT: Dr. Carlos Blazquez, Univ. of Florida Citrus Data Center, Lake Alfred, FL (813) 956-1151

NEEDS: To establish a method of storage and retrieval for large quantities of aircraft data.

SOLUTION: Use an expert system front-end linked to an analog laser optics player for image retrieval.

STATUS: Discussion begun with Dr. Blazquez to understand requirements. Dr. Blazquez has visited UCF Intelligent Multimedia Applications Laboratory (IMAL) for demonstrations. Further discussion are required but unscheduled.

* *

APPLICATION # 7: NASA-KSC Facility Drawing Retrieval

ORGANIZATION: EG & G Engineering Records

NASA CONTACT: Dave Springer, 867-3035

USER CONTACT: TBD

NEEDS: Approximately TBD facility drawings are stored. Drawings presently stored on microfisch for later reproduction on hard copy. User is required to know drawing number for retrieval.

MEDIA: Microfisch drawings.

SOLUTION: Use an expert system front-end linked to an analog laser optics player for image retrieval.

STATUS: No action taken for this potential application.

* *

APPLICATION # 8: Payload Customer Integration

ORGANIZATION: NASA-KSC, CP-APO

NASA CONTACT: Robert Young, CP-APO, 867-3374

USER CONTACT: Beth Cerrato, CP-PSO-A, 867-3183; Shirley Green CP-APO, 867-3374

NEEDS: NASA and their contractor representatives need a method to assist in evaluating commercial space experiment KSC processing requirements.

MEDIA: Text, database, possibly motion and still imagery.

SOLUTION: Use and expert systems front-end linked to a text and image database.

STATUS: Three (3) College of Business MBA students have developed a prototype systems during the Fall 1990 semester. In addition, one (1) College of Engineering student is developing a payload assignment/tracking systems database during the Fall 1990 and Spring 1991 semesters.

* *

APPLICATION # 9: Bionetics Environmental Aircraft Image Search and Retrieval System

ORGANIZATION: Bionetics

NASA CONTACT: Dr. William Knott, MD-RES-L, 853-5142

USER CONTACT: Dr. C. Ross Hinkle, Bio-2, 853-3281

NEEDS: To establish a search and retrieval system for approximately TBD 10" x 12" aircraft photographs.

MEDIA: Aircraft photography.

SOLUTION: Use an expert system front-end linked to an analog laser optics player for image retrieval.

STATUS: No actions taken for this potential application.

* *

APPLICATION # 10: NASA Safety Reliability & Quality Control Photo Retrieval System

ORGANIZATION: NASA-KSC SR & QA

NASA CONTACT: Judy Anderson Kersey, 867-3163

USER CONTACT: TBD

NEEDS: Develop a method for storage, search, and retrieval for a variety of R&QA photographs.

MEDIA: Various sized photographs.

SOLUTION: Use an expert system front-end linked to a laser optics device for data retrieval.

STATUS: No action taken for this potential action.

* *

APPLICATION # 11: Shuttle Processing As-Run Shuttle Historical Data System

ORGANIZATION: NASA-KSC

NASA CONTACT: Dick Thornburg/Gene Sestile, TM-PCO-2, 867-7401/-3844

USER CONTACT: As above.

NEEDS: A system which would provide quick access to shuttle processing as-run historical data.

MEDIA: Hard copy graphical/scheduling data.

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: No action taken for this potential action.

* *

APPLICATION # 12: Safety, Fire, and Security Facility Training System

ORGANIZATION: Safety and Protective Services

NASA CONTACT: Jim Lansing, RQ-SS, 867-4802

USER CONTACT: TBD

NEEDS: Several organizations within safety and protective services have periodic requirements for "surrogate travel" training systems for facility familiarization.

MEDIA: Motion, still, computer generated graphics, and text.

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: No action taken for this potential action.

* *

APPLICATION # 13: Spaceport USA Interactive Exhibits

ORGANIZATION: NASA

NASA CONTACT: Ed Harrison, PA-VIC, 867-2363

USER CONTACT: TBD

NEEDS: There is a continual need for interactive exhibits for Spaceport USA visitors.

MEDIA: Computer interfaced laser disc, CD-ROM, DVI, etc.

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: No action taken for this potential action.

* *

APPLICATION # 14: Expert Maintenance Diagnostics/Multimedia

ORGANIZATION: General Dynamics Space Systems (GDSS)

NASA CONTACT: None

USER CONTACT: Rick Ort, GDSS, 853-2793

NEEDS: Various complex launch complex systems exist which require specialized experience for diagnostics during a period where experienced people are retiring. Multimedia ties to a knowledge-based system would be invaluable for the retrieval of schematics, still photographs, and motion sequences.

MEDIA: Still video, motion, and drawings.

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: Three (3) College of Business MBA students have developed a prototype diagnostics multimedia system during the Fall 1990 semester.

* *

APPLICATION # 15: Equipment Trouble Shooting Expert

ORGANIZATION: ITT Defense Systems

NASA CONTACT: None

USER CONTACT: Ken Shaw, ITT, 633-7753

NEEDS: Diagnostic trouble shooting systems are needed to verify equipment performance and acceptance testing.

MEDIA: Still video, motion, and drawings.

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: One (1) College of Business MBA student has developed a prototype diagnostics system during the Fall 1990 semester.

* *

APPLICATION # 16: Multimedia Selection Advisor

ORGANIZATION: UCF College of Business

NASA CONTACT: None

USER CONTACT: Dr. James Ragusa, UCF, 823-2915

NEEDS: Various organizations need methods of determining the best media to use in presenting a specific message. In addition, those interested in Executive Information Systems (EIS) need rapid access to a variety of presentation multimedia.

MEDIA: Varied

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: Three (3) College of Business MBA students have developed a prototype multimedia system during the Fall 1990 semester.

* *

APPLICATION # 17: Intelligent Computer Aided Training/ Instruction

ORGANIZATION: NASA, DOD, NSF, and others

NASA CONTACT: None

USER CONTACT: Dr. James Ragusa, UCF, 723-2915

NEEDS: There is an ever expanding need for one-on-one,

self-paced intelligent Computer Aided Training/ Instruction. Strong interest by military training organizations.

MEDIA: Varied

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: Two (2) College of Business MBA students have developed a prototype multimedia system during the Fall 1990 semester. Two (2) MBA students will continue this development during the Spring 1991 semester.

* *

APPLICATION # 18: Quality Control Training

ORGANIZATION: NASA and others

NASA CONTACT: None

USER CONTACT: Dr. James Ragusa, UCF, 823-2915

NEEDS: Need exists for all levels of interactive quality training from solder certification to total quality control.

MEDIA: Varied

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: No action taken for this potential action.

* *

APPLICATION # 19: Training Devices Costing Estimator

ORGANIZATION: U.S. Army (PM TRADE)

NASA CONTACT: None

USER CONTACT: Richard Insinger, U.S. Army (PM-TRADE), 380-4349

NEEDS: Organizations responsible for the development of training devices need accurate methods of determining device costs based on requirements, specifications, and historical development experiences.

MEDIA: Varied

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: Three (3) College of Business MBA students have developed a prototype system during the Fall 1990 semester.

* *

APPLICATION # 20: Business Ethics Advisor

ORGANIZATION: UCF and Arthur Anderson & Co

NASA CONTACT: None

USER CONTACT: Dr. James Ragusa, UCF, 823-2915

NEEDS: There is a need in our contemporary society for providing quality advice concerning ethics issues. Sources of this advice can be based on norms, laws, expectations, and policies.

MEDIA: Varied

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: Three (3) College of Business MBA students have developed a prototype system during the Fall 1990 semester. This work was supported by Arthur Anderson & Co.

* *

APPLICATION # 21: Customer Help Advisor

ORGANIZATION: TBD

NASA CONTACT: None

USER CONTACT: Dr. James Ragusa, UCF, 823-2915

NEEDS: Providing quality help and advice at a service decision point is needed in numerous organizations and situations.

MEDIA: Varied

SOLUTION: Use an expert system front-end linked to various forms of laser optics for data retrieval.

STATUS: No action taken for this potential action.

S2-82
333522
N91-70700
p.73

NASA KSC

Intelligent Interactive High Speed Data Search System

11 - 28 - 203

Annual Report
1990

Principal Investigator

Dr. James R. Driscoll

Associate Professor
College of Arts & Sciences
Department of Computer Science
University of Central Florida

(407) 823-2341

TABLE OF CONTENTS

Section 1.

Overview

C-2

The overall objective of this three year Cooperative Agreement project is to research a system capable of high speed search of very large text databases in order to (1) produce answers to questions expressed in natural language, and (2) generate summaries of database information. The objectives listed for Year 1 of this project are to essentially prepare for item (1) above. A somewhat detailed, but readable, three page description of item (1) appears in Section 2 of this report. During Year 1, this three page description was submitted to NASA Headquarters as a KSC project white paper for the Advanced Operations Program managed by Chuck Holliman.

The objectives for Year 1 have been achieved. However, there has been a crucial modification. Very simply, and with slight rewording from the Cooperative Agreement, the Year 1 objectives have been the following:

a. Design an internal format for analyzed text.

b. Develop a prototype text parsing system in regard to the format.

c. Establish the "dictionary" required for the prototype text parsing system.

d. Study the structure of English questions.

e. Purchase required PC platforms.

f. Put in place a high speed basic (shell) text retrieval system in preparation for extension to answering questions.

The crucial modification mentioned earlier affects objectives (a), (b), and (c). In order to extend our work beyond the performance of a prototype and to make our work applicable (without modification) to all text, we have eliminated the need (for now) to do precise parsing of English sentences and questions. Instead, our modification is to make use of basic, general world knowledge and improve well-known text retrieval techniques to accomplish our objective of being able to generate an answer to an English question.

A description of the modification affecting objectives (a), (b), and (c) can be found in Section 3 of this report. We are just now beginning to implement the procedures identified there. Preliminary experiments have shown good results. This implementation project will yield a Master's degree for Edgar Wendlandt.

Section 4 of this report is aimed at objective (c). Described there is a proposal which has recently been funded by the State of Florida High Technology and Industry Council (FHTIC). The intent of the funding is to develop a "dictionary" of general world knowledge and carry our research beyond the prototype stage and into commercialization. This funding begins in January 1991. The proposal was submitted in August of 1990 by Dr. Driscoll.

Section 5 presents an analysis of English question structure in regard to the general world knowledge aspect of our research. The report is somewhat technical and really not intended for a general audience but is included here solely to provide evidence of work on objective (d) of Year 1. This report was essentially generated by undergraduate student Gloria Corbett as an independent study project. No NASA funds were used to support this activity.

In regard to objectives (e) and (f), we have purchased two 33 MHz 386 PC clones, each with a 165 Mbyte, 14.5 ms hard drive. We have developed a high speed basic text retrieval system, and recently installed the system on one of these two machines. The retrieval system was implemented in the C programming language by three undergraduate students: Frank Fenneran, Bill Gellerstedt, and Mike Cuccarese. This activity was done as a classroom project and under several independent study agreements. No NASA funds were used to support this activity. The system is modeled after the French commercial product SPIRIT which is described in Section 6 of this report where a recent publication by Dr. Driscoll is provided.

Finally, to conclude this overview of project progress during Year 1 of the Cooperative Agreement, the following is a list of presentations during 1990 where progress reports were submitted to KSC.

Purpose	Date	Location
1. Briefing	20 February 1990	UCF
2. Project Review	20 March 1990	UCF
3. Video-Teleconference	23-24 April 1990	NASA Headquarters, JSC, KSC
4. Project Review	20 July 1990	KSC
5. Project Review	19 October 1990	UCF

Section 2.

White Paper Description of Project submitted to NASA Headquarters for the Advanced Operations Program

INTELLIGENT INTERACTIVE HIGH SPEED DATA SEARCH SYSTEM

KSC

James R. Driscoll

Department of Computer Science
University of Central Florida
P.O. Box 25000, Orlando, FL 32816
(407) 275-2341

This is a new project. It concerns natural language access to large amounts of textual information. The objective is to automate the access of textual information so that users are not overwhelmed by the amount of information searched. Currently, the project is motivated by the desire to provide convenient access to information contained in the numerous and large public information documents maintained by Public Affairs at KSC.

The documents maintained by Public Affairs at KSC consist of press releases and other printed information created at KSC and other NASA offices using various word processors. There are also documents from outside contractors, such as Rockwell, which produces the "NASA National Space Transportation System Reference" more often called the "shuttle manual". During a launch at KSC, about a dozen NASA employees access these printed documents to answer media questions. The amount of text is visually overwhelming. The shuttle manual alone is just over 1,000 pages (it is three inches thick). The planned document storage for NASA KSC Public Affairs is around 300,000 pages (approximately seventy-five feet of stacked pages) with about 5,000 pages added or replaced annually.

In regard to the operation of the Public Affairs Office at KSC, the following concerns have been identified and are addresed by this project:

a. The process of obtaining information from large volumes of printed documentation is manual; it is cumbersome and not fast enough. An increase in media questions is expected.

b. There is a turnover in personnel and long training periods are required to learn how to properly answer media questions.

c. Answers to media questions must be saved and later searched to avoid different answers to the same question and to reduce duplication of effort.

d. An increase in the amount of documentation is expected.

e. Coordination with the Public Affairs operations within other NASA space centers is anticipated.

To summarize, the aim is to eliminate an employee turnover and training problem for a regularly performed, publicly visible task which is tedious, labor intensive, and can produce an inconsistent result.

Progress to date has been to research hardware and software requirements for a prototype intelligent interactive high speed data search system to automate the manner in which media questions are answered. The effort has led to a prototype design aimed at expository text (technical manuals, reports, regulations, research papers, etc.). The prototype takes advantage of IR techniques to find relevant text, and it uses general world knowledge to answer queries and summarize text. As diagrammed in Figure 1, it is characterized by the following three activities:

I. Automatic indexing based on syntactic, linguistic, and probabilistic techniques to retrieve a ranked list of relevant text for a given natural language query.

II. Converting retrieved relevant text to a conceptual format based on general or world knowledge.

III. Letting meaning be represented by "portions" of the original text and using portions of retrieved text to build a response to a query or generate a summary.

This design concerns natural language processing and involves the acquisition of general world knowledge. These are complex tasks still open to research and criticism. Consequently, some detail is necessary to make it clear as to what will be accomplished by this project.

Activity I is not new; there are commercial systems which already perform this activity. These systems are complex but rather straightforward to implement. They are based on frequency of occurrence of character strings and the more sophisticated systems know how words should be spelled and what part of speech a word can have (e.g., noun, verb, adjective, etc.). Referring to Figure 1, for a demonstration, the 1,000 page shuttle manual (stored electronically as a three megabyte text file) was used by considering each paragraph of the manual as a document. This resulted in a collection of 4,902 documents. A commercial hypertext system called SPIRIT was used to automatically index the collection and provide natural language access. For the natural language query "When does NASA select astronaut candidates?", a ranked list of forty-three relevant paragraphs was retrieved. Paragraph number 3386 was the most relevant, paragraph number 2132 was the second most relevant, etc. For the prototype system, a ranked list of paragraphs retrieved for a query is stored in RAM to facilitate further processing.

Activity II is based on a linguistic concept called thematic roles. Thematic roles help question answering by revealing how sentence phrases and clauses are related to the verbs and modifiers in a sentence. It is easy to explain thematic roles by using an example and avoiding some detail. Consider the following sentence:

Mary made coffee for John with a percolator.

There are four noun phrases in this sentence, each of

which fits into a particular thematic role as follows:

Noun Phrase	Thematic Role
Mary	AGENT
coffee	THEME
for John	BENEFICIARY
with a percolator	INSTRUMENT

Four corresponding questions can be answered now:

What was made? -> THEME -> coffee
Who made coffee? -> AGENT -> Mary
For whom was coffee made? -> BENEFICIARY -> John
With what was coffee made? -> INSTRUMENT -> a percolator

There are approximately thirty known thematic roles.

It is important to point out that this project is not concerned with figuring out, for example, what "a percolator" is; we only need to know that this string of characters occupies an INSTRUMENT position in the thematic form of a sentence. This is perhaps the most important point behind the success of this project.

In order to convert text to thematic form, the following general world knowledge is needed:

a. Certain words suggest specific thematic roles. For example, all verbs other than the "be-verbs" (is, are, was, etc.) become candidates for an ACTION. As another example, the word "noon" implies TIME.

b. Prepositions and certain conjunctions trigger thematic role possibilities. For example, the preposition "for" can trigger the BENEFICIARY or DURATION thematic roles. The conjunction "as" can possibly trigger the MANNER or TIME thematic roles.

c. Normally, verbs cause thematic grids of the form AGENT followed by ACTION as in

   John ran.

   or the form AGENT followed by ACTION followed by THEME as in

   John chased the dog.

   Each of these grids can then be followed by some of the other thematic roles as in

   John chased the dog into the house.

   Here, "into the house" is a DESTINATION thematic role. But there are other thematic grids. For example, there is a class of verbs (represented by the verbs "load" or "spray") which can have a grid which specifies an AGENT, ACTION, LOCATION, THEME order, as in

   John loaded the truck with furniture.

   Here, "the truck" is a LOCATION rather than a THEME.

The conversion process uses the knowledge outlined above for all words which may appear in sentences to be converted. For a somewhat robust system, this would involve approximately 100,000 words. Items B amd C are the most important with item C being the most time consuming to establish. The conversion process can operate even if knowledge for items A and C is not known for domain specific words; this makes the process domain independent.

Acquisition of the above knowledge will be done manually from memory, but also from reading English dictionary entries when we are not so sure. Creating this knowledge is a time consuming process. However, it is felt that the knowledge can be automatically produced directly from standard English dictionary text. We have only just begun to consider procedures for this task. In any case, the knowledge must be established and it can be obtained from an English dictionary. This is shown as an input to Activity II in Figure 1 and labeled as mass storage for common knowledge.

The output of Activity II is a collection of thematically parsed sentences stored in RAM to facilitate further processing. Referring to Figure 1, a thematic parse for the sentence "Astronaut candidates are selected as needed for training at JSC." is shown as an example. This sentence occurred in paragraph number 4011 of the shuttle manual, the third most relevant paragraph for the query "When does NASA select astronaut candidates?". For the forty-three retrieved paragraphs shown as input to Activity II, there are 160 sentences and potentially each one would have to be converted to thematic form.

Activity III is rather straightforward. An answer to the query is generated by a scan of parsed sentences for one having a TIME portion, where the ACTION specified is "select" and the THEME specified is "candidates"; and if an AGENT is present, it must have the value "NASA". The response is the TIME portion of such a sentence. In Figure 1, this is shown as the phrase "AS NEEDED" on the computer monitor, taken from the sentence mentioned above and diagrammed on the RAM device in Figure 1.

A crude version of the SPIRIT system's retrieval capabilities has been implemented in the C programming language for a PC or SUN workstation. Speed is not a problem for this activity; good performance can be achieved in a PC environment. A crude thematic parser is under development now to investigate speed performance in a PC environment for Activities II and III. The Prolog programming language is being used to enable rapid development of the parser. Parallel parsing techniques are also being considered because high speed appears to be an issue here.

The project's schedule includes a quickly implemented prototype incorporating all three activities by October 1990. This system will only have a 600 word vocabulary but should demonstrate relevant text retrieval and thematic parsing ability. Its purpose is to investigate the speed issue mentioned for Activity II. The current level of funding allows for manual development of a 14,000 word vocabulary and demonstration of a question/answer prototype in December 1991, and a summarization prototype in December 1992. These prototypes could handle, for example, the shuttle manual.

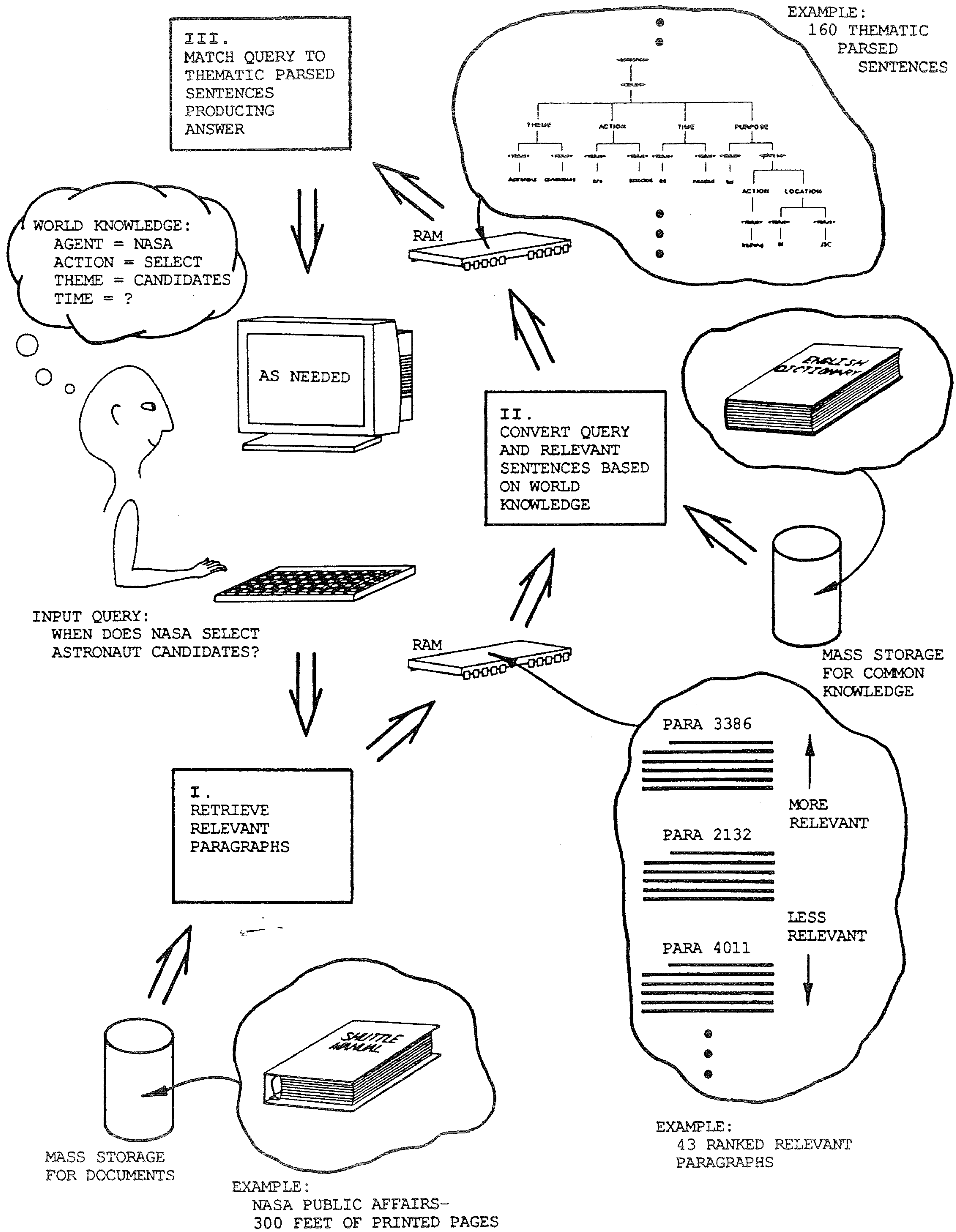


Figure 1: Three Activities for Prototype Question/Answer System.

Section 3.

Master's Degree Research Project Proposal submitted to Graduate Committee, Department of Computer Science, UCF

IMPLEMENTATION OF A THEMATIC ROLE AND ENTITY ATTRIBUTE LEXICON FOR INFORMATION RETRIEVAL FROM FULL-TEXT DATABASES

Research Project Proposal Submitted to:

Graduate Committee
Department of Computer Science
University of Central Florida

by

Edgar B. Wendlandt

Advisory Committee

Dr. J. R. Driscoll, Chairman
Dr. M. A. Bassiouni
Dr. S. D. Lang

November 13, 1990

I. Introduction

Full-text information retrieval (IR) systems differ from record based IR systems. Identifiers, also referred to as keywords, index terms, or descriptors, are attached upon entry of the text data . The identifiers may be attached manually or they may be automatically assigned. The automated process performs two functions of interest during the initial entry of the data. First, it throws out useless or empty words. These words are generally prepositions. Second, it calculates weighting factors. The weighting factors indicate how useful a document is to a query. Document size is variable and generally assigned by the user. For instance, each paragraph in a text may be considered a document. The calculation of the weighting factor *(w)* is a combination of term frequency *(tf)*, document frequency *(df)*, and inverse document frequency *(idf)*. The terms are defined as follows:

tf_{ij} = number of occurrences of term T_j in document D_i

df_j = number of documents in a collection which contain T_j

idf_j = $\log(N/df_j)$; where N = total number of documents.

w_{ij} = $tf_{ij} * idf_{ij}$

When the system is queried it computes a vector Q of weighting factors. The retrieval of a document is based on the value of a similarity coefficient. In current literature the degree of similarity is most often a factor of term frequencies, co-occurrence relationships, and proximal distances [DEB89,RAU89,ZER90]. Some common similarity factors are calculated as follows:

$$sim_1(Q,D_i) = \sum_{j=1}^{t} w_{qj} * d_{ij}$$

$$sim_2(Q,D_i) = \frac{\sum_{j=1}^{t} w_{qj} * d_{ij}}{\sqrt{\sum_{j=1}^{t} d_{ij}^2}}$$

$$sim_3(Q,D_i) = \frac{\sum_{j=1}^{t} w_{qj} * d_{ij}}{\sqrt{\sum_{j=1}^{t} d_{ij}^2 * \sum_{j=1}^{t} w_{qj}^2}}$$

The function sim_1 is a crude but simple coefficient. The functions sim_2 and sim_3 normalize the coefficients in case of different document sizes.

II. Objective

The objective of the project is to modify and potentially improve an existing IR system. The existing system is now being applied to Dr. Driscoll's NASA KSC funded research. The system currently uses inverted indexing and keyword weighting as described. In addition it has the ability to recognize synonyms of keywords.

As an example, given a query about the space shuttle program, the system currently aids the user in obtaining relevant documents, which are paragraphs from the space shuttle manual (a one-thousand page reference manual). Output from the current system is generally a long list of rank ordered paragraph numbers. The user must then scan these paragraphs for the factual information desired. Refer to Figure 1 where a sample query is shown at the top. The system returns a list of six empty words, seven keywords, and a listing of relevant paragraph classes. These classes are groups of paragraphs of similar weight where the number of paragraphs in a group is listed under NB DOCS. The keywords that triggered the retrieval of these paragraphs is listed to the right. The class of paragraphs are listed from most relevant to least relevant as determined by the present system. The only paragraph from class one is shown at the bottom of the figure. In this case, the answer to the query happens to be in that paragraph.

Refer to Figure 2, where the same system was used to answer 35 questions. (Appendix A provides a list of the questions.) The figure shows that it is not always the case that the first paragraph contains the answer to the question. In the figure, the right most column shows the number of paragraphs that were scanned in the order the system ranked them before the answer was found. The last line indicates that an average of three paragraphs had to be scanned to find the answer to the questions but notice that in a few cases, eleven paragraphs were scanned before the answer was found.

Shuttle Data Base: 5000 documents, each a paragraph in the shuttle manual

<1> : How long does the payload crew go through training before a launch?

EMPTY WORDS: how, does, the, through, before, a.
KEY WORDS: long, does, payload, crew, go, training, launch.

CLASSES	NB DOCS	KEY WORDS
1	1	Payload-crew, training, launch
2	2	payload-crew, training
3	1	long, payload, crew, training, launch
4	1	long, payload, crew, training
5	1	payload, crew, go, launch
6	1	does, go, launch
7	1	does, payload, training
8	1	does, crew, go
9	1	crew, go, training
10	1	long, payload, training
11	1	long, does, crew
12	1	does, payload, launch
13	3	payload, training, launch
14	2	crew, go, launch
15	1	does, crew, launch
16	2	crew, training, launch
17	3	payload, crew, go
18	17	payload, crew, training
19	2	long, crew, launch

Ignored Thematic Information:
1. Query
   - How long ⇒ DURATION, ...
   - through ⇒ DURATION
   - before ⇒ TIME
2. Sentence Answer
   - through ⇒ DURATION
   - before ⇒ TIME

DOC 3386 BASE:doc 3386 NCP:0/CPI:1/NBI:1 +18 1K/1K
IDENTIFIER:doc 3386
TEXT.........:
Training requirements depend on the complexity of both the instruments and the integrated payload. The investigator helps determine training requirements for his instrument and participates in training the payload specialists and support personnel who may operate equipment, monitor data or assist in troubleshooting. Investigators who participate in such activities also require indoctrination and training in practices and equipment operation. Payload crew training currently begins 1.5 to two years before launch and continues through launch. Training on individual experiments and experiment instruments is normally scheduled before payload integration, which begins approximately one year before launch.

Figure 1. Sample Query

Results of finding answers using SPIRIT:
(ordered by question number)

	Answer found?	Search Time	Number of documents read
1)	found	1 minute 45 seconds	1
2)	found	1 minute 45 seconds	1
3)	found	1 minute 45 seconds	2
4)	found	2 minutes 0 seconds	3
5)	found	1 minute 30 seconds	2
6)	found	2 minutes 0 seconds	3
7)	found	1 minute 30 seconds	1
8)	found	3 minute 0 seconds	3
9)	found	1 minute 15 seconds	1
10)	found	1 minute 15 seconds	1
11)	found	4 minutes 30 seconds	11
12)	found	1 minute 0 seconds	1
13)	found	1 minute 0 seconds	1
14)	found	2 minutes 45 seconds	6
15)	found	2 minutes 30 seconds	6
16)	found	1 minute 0 seconds	1
17)	found	1 minute 30 seconds	1
18)	found	1 minute 0 seconds	1
19)	found	2 minutes 0 seconds	2
20)	found	2 minutes 15 seconds	5
21)	found	1 minute 0 seconds	1
22)	found	2 minutes 0 seconds	1
23)	found	3 minutes 0 seconds	5
24)	found	1 minute 30 seconds	1
25)	found	3 minutes 30 seconds	11
26)	found	3 minutes 45 seconds	8
27)	found	1 minute 15 seconds	1
28)	found	2 minutes 0 seconds	2
29)	found	3 minutes 0 seconds	4
30)	found	2 minutes 45 seconds	6
31)	found	3 minutes 30 seconds	5
32)	found	1 minute 15 seconds	1
33)	found	1 minute 45 seconds	1
34)	found	1 minute 15 seconds	1
35)	not found	1 minute 0 seconds	0*

*Could determine in one minute that answer was not in the book. Notice that the word "color" was not found as a keyword.

Average search time = 2 minutes 2 seconds

Average number of documents read = 3

Figure 2. Query Results

It has been suggested that thematic roles and entity attributes may be used to better extract the desired document [SAL89,ZER90]. Thematic roles are a linguistic concept. They could help in question answering by revealing how sentence phrases and clauses are related to the verbs and modifiers in a sentence. There are approximately thirty known thematic roles. Entity attributes are generally considered descriptors of something. They are important in answering questions about an object's attributes. For example, color and weight are considered attribute categories.

In this respect, the current system throws out some useful information when it blindly throws out the empty words. Again, refer to Figure 1 where the box out to the right shows ignored thematic information. The box shows that the thematic information of duration and time were ignored when the words "how," "through," and "before" were thrown out as empty words. Also, it shows that the sentence containing the answer has thematic information of duration and time.

A major obstacle in the use of thematic roles and entity attributes is the non-availability of an appropriate lexicon [ZER90]. Therefore, a thematic role and entity attribute lexicon must be built. The goal is to implement a system that uses the lexicon to better place the most relevant paragraphs relating to a query at the head of the list.

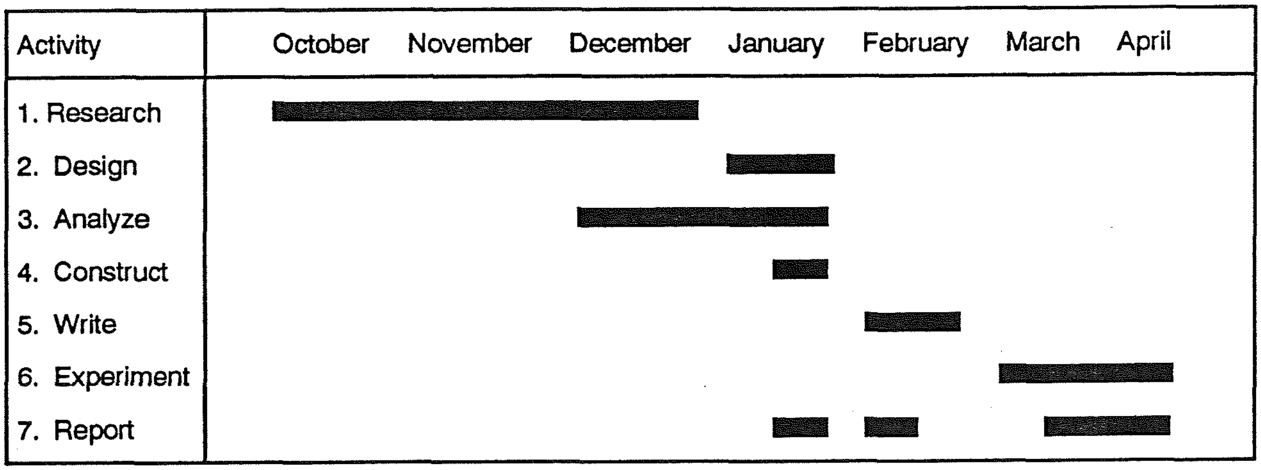
III. Activities and Significance

The first portion of the project requires designing and manually building a limited lexicon for a text file containing 26 pages of the NASA Shuttle Manual. After the lexicon has been designed and constructed, the existing IR system will be modified to incorporate the use of thematic roles and attributes. Weighting factors for thematic role and entity attribute occurrence will be computed in a fashion similar to the weighting factors mentioned in Section I. This new weighting factor will be combined with the existing weighting factor in similarity coefficient calculations. Details of the combination will be determined as an activity for this project.

In the second phase of the project, a series of experiments will be run. The output of the modified system will be compared to that of the original system. Depending on the results of the initial experiments the newly introduced weighting factors may be proportioned to place more or less significance on the thematic roles and attribute characteristics.

Finally, the results of the experimentation phase will be evaluated. The evaluation will provide some feedback about the lexicon's effectiveness and content. This feedback may aid in developing algorithms for general lexicon construction. Also, it should indicate whether or not the concept of thematic roles is significant in the area of fact retrieval from full-text databases.

IV. Chronology



1. Research of current techniques used in IR and submit proposal.
2. Design the lexicon and algorithms for incorporating its use (4 weeks).
3. Analyze the source code of the current IR system (10 weeks).
4. Construct a limited domain lexicon (2 weeks).
5. Write and implement the source code modifications (4 weeks).
6. Experiment, result analysis (8 weeks).
7. Report Preparation.

V. Detailed Chronology

Activity 1: The research has been ongoing throughout the fall semester.

Activity 2: The design of the lexicon will require specifying how the lexicon will be stored on the computer. The algorithm design will specify how the lexical information will be incorporated.

Activity 3: The IR system to be modified is a C program written by students under the guidance of Dr. Driscoll. I need to determine how and where functions may be added to modify the weighting to reflect the addition of lexical information.

Activity 4: The building of the lexicon will require that I manually scan the test document noting words with thematic and attribute information. Also, a set of words "typically" used in queries will need to be determined. These words and their associated thematic and attribute information will also need to be included in the lexicon.

Activity 5: The C source code will need to be written for the functions designed in Activity 2. The current IR system source will also be modified to incorporate these functions as determined in Activity 3.

Activity 6: The experiment will require running the original system and the new system on some test queries. The results will be analyzed to determine if the thematic and attribute information was incorporated as best as possible and if not modifications may be made requiring additional runs.

Activity 7: The report preparation will be ongoing. It will include a summation of how the lexicon was constructed, a summation of exactly how the thematic and attribute information was incorporated, an analysis of the results, and conclusions.

VI. List of Deliverables

1. Design document and the algorithms for incorporating lexicon.
2. Limited Domain Lexicon.
3. Design document and the source listing for the IR system modification routines.
4. Experiment results.
5. Project report.

VII. BIBLIOGRAPHY

[DEB89] Debili, Fathi, "About Reformulation in Full-Text IRS," Information Processing and Management (Vol 25), 1989, pp. 647-657.

[KEM88] Kemp, Alasdair D., Computer-Based Knowledge Retrieval, The Association for Information Management, 1988, pp. 4-1 - 4-29.

[RAU88] Rau, Lisa F., "Conceptual Information Extraction and Retrieval From Natural Language Input," RAIO (Vol 1), 1988, pp. 424-437.

[RAU87] Rau, Lisa F., "Knowledge Organization and Access in a Conceptual Information System," Information Processing & Management (Vol 23 No 4), 1987, pp. 269-283.

[RAU 89] Rau, Lisa F., Paul S. Jacobs, Uri Zernik, "Information Extraction and Text Summarization Using Linguistic Knowledge Acquisition," Information Processing & Management (Vol 25 No 4), 1989, pp. 419-428.

[SAL89] Salton, Gerard, Automatic Text Processing, Addison Wesley, 1989, pp. 275-375.

[ZER90] Zernik, Uri, and Paul Jacobs, "Tagging for Learning: Collecting Thematic Relations from Corpus," COLING-90 (Vol 1), 1990, pp. 34-39.

Appendix A

Shuttle Questions:

1) What is the maximum cargo weight the shuttle can carry?
2) How far can the shuttle transport cargo from the earth's surface?
3) What has happened to the Enterprise?
4) How many years of education are required for astronaut candidates?
5) What is the total weight of the shuttle?
6) How thick is the window of the shuttle?
7) How many gallons of liquid hydrogen fuel can the storage tank hold?
8) What type of liquid fuel is used on the shuttle?
9) What is the descent rate of the shuttle during landing?
10) How long is the mechanical arm used for payload deployment?
11) What are the dimensions of the cargo area in the shuttle?
12) How is waste disposed of?
13) Have there been astronauts picked from minority groups?
14) What is the total number of times that the shuttle has been launched?
15) What type of food do astronauts eat during a shuttle mission?
16) What is the orbiter's velocity while in orbit?
17) What is the maximum acceleration of the shuttle during launch?
18) What is the maximum touchdown glide speed of the shuttle?
19) How many pounds of thrust do the SRB booster rockets generate during liftoff?
20) What is the maximum fluid fuel flow rate during launch?
21) How fast does the crawler or transporter travel?
22) At what altitude and speed must the pilot throttle back during ascent?
23) How many general purpose computers are on board the shuttle and what functions do they serve?
24) What is the new design of general purpose computers like on board the shuttle?
25) What is the total number of tiles that cover the orbiter for thermal protection during reentry?
26) When did the first space shuttle launch occur?
27) How long is the runway at a shuttle landing facility?
28) What type of glass is used for the windows to withstand the pressure of flight?
29) What is the total amount of RAM available in the shuttle's general purpose flight computer?
30) What material are the heat shield tiles composed of?
31) What type of computer guidance and navigation system does the shuttle use during reentry and landing?
32) What is the maximum power available to the payload area?
33) Are there emergency escape procedures to jettison the crew members out of the shuttle?
34) Where do the crew members sleep on the shuttle?
35) What is the color of the external tank? (no answer)

Section 4.

Proposal funded by State of Florida,

High Technology and Industry Council (FHTIC)

(Note to SUS DSR, this is top page of submitted proposal)

A Proposal
submitted to

State of Florida
HIGH TECHNOLOGY AND INDUSTRY COUNCIL
APPLIED RESEARCH GRANTS PROGRAM

by

UNIVERSITY OF CENTRAL FLORIDA
Division of Sponsored Research
P. O. Box 25000 ADM 243
Orlando, Florida 32816-0001
(407) 275-2671

Title: Intelligent Text Processing

Planning: X Operational: ______ No. of years funded to date: 0

Technology Area: Software and Computer Science

Principal Investigator(s):

Name: James R. Driscoll

Signature: J. Driscoll

Telephone: (407) 275-2341

Soc. Sec. No. [redacted]

Dept. or Unit Affiliation: Dept. of Computer Science

Institution (for consortium members):

Proposal: New X Renewal ______ 1990-91 Amount Requested: $ 19,986.00

Executive Summary (100 words): This proposal builds upon ongoing externally funded reserach in regard to development of a prototype, PC-based, intelligent high speed text data search system. This proposal has two objectives. The first objective is to develop a fully-functional, PC-based text retrieval system incorporating the very latest proven linguistic techniques. We expect this first objective to yield a commercial product. The second objective is to plan for commercialization of more intelligent text processing systems for such tasks as answering questions, writing summaries, correcting grammar, or translating text from one language to another.

Endorsements:

	Department Head (or Dept. Center Director)	College Official (or Multi-Dept. Center Director)	University Official
Name	Dr. Ron Dutton	Dr. Bruce A. Whisler	Dr. Joan R. Burr
Signature	Ronald Dutton	Bruce A Whisler	Joan R. Burr
Title	Assoc. Chairman	Assistant Dean for Budget	Director, Division of Sponsored Research
Telephone			
Date			9/10/90

TABLE OF CONTENTS

INTRODUCTION

Everyone is familiar with the commercial success of word processors and their presence on virtually every desk in an office environment. These systems are invaluable for creating documents, and often appear intelligent when they perform tasks such as correcting spelling. Now, an era of commercially available text processing programs is approaching. For example, we can expect to see, in the not too distant future, intelligent text processors which can correct grammar, search existing text and answer questions, scan existing text and write summaries, and automatically translate text to other languages.

It is reasonable to expect these products to be just as prevalent as word processors are today. Several factors are contributing to this expectation. One is the advance in storage and computing power of "desktop" computers. Another is the ability to store and process large machine readable dictionaries and thesauri. And a third is the ability to store and process massive corpora of "everyday" text.

These advances and abilities have triggered new linguistic research. Automatically parsing sentences, automatically obtaining semantic information from dictionaries, and machine translation were the most prevailing research topics presented at the 13th International Conference on Computational Linguistics held August 16-25, 1990. In addition, intelligent text handling has become a rapidly developing research field in its own right. Refer to the Call for Papers and Call for Demonstrations at the RIAO 91 Conference on Intelligent Text and Image Handling in ADDITIONAL SUPPORTING DOCUMENTATION.

Still, there are some significant research hurdles for intelligent text processing. The TECHNICAL DISCUSSION which follows describes a project at the University of Central Florida (UCF) for developing a prototype system for natural language access to large amounts of text. The research has been supported by DoD Army Research Institute/PM-Trade Contract

N61339-88G-0002 Order 0004,[1] and by NASA Kennedy Space Center (KSC) Grant NAG 10-0058 Project 2A.[2] This ongoing project at UCF attempts to solve a major text processing research hurdle and several facets of this funded research could lead to commercialization.

TECHNICAL DISCUSSION

1. Detailed Statement of the Problem

This project concerns natural language access to large amounts of text. The objective is to automate the access of textual information so that users are not overwhelmed by the amount of information searched. Currently, the project is motivated by the desire to provide convenient access to information contained in the numerous and large public information documents maintained by Public Affairs at KSC.

The documents maintained by Public Affairs at KSC consist of press releases and other printed information created at KSC and other NASA offices using various word processors. There are also documents from outside contractors, such as Rockwell, which produces the "NASA National Space Transportation System Reference" more often called the "shuttle manual." During a launch at KSC, about a dozen NASA employees access these printed documents to answer media questions. The amount of text is visually overwhelming. The shuttle manual alone is just over 1,000 pages (it is three inches thick). The planned document storage for NASA KSC Public Affairs is about 100 million words of text or around 300,000 pages (approximately seventy-five feet of stacked pages) with about 5,000 pages added or replaced annually. Electronically, the storage is manageable; approximating 900 megabytes of disk space is expected.

---

1 This research demonstrated the usefulness of thematic roles (surface knowledge) in automated diagnostic systems. Thematic roles were used to match a user's description of a problem with symptoms, to obtain procedures to correct the problem.

2 This research provided a survey of hardware and software technology appropriate to automating the manner in which NASA KSC Public Affairs personnel obtained answers to media questions. It resulted in a demonstration of a commercial state-of-the-art text retrieval system to NASA KSC.

In regard to the operation of the Public Affairs Office at KSC, the following concerns have been identified and are addressed by this project:

a. The process of obtaining information from large volumes of printed documentation is manual; it is cumbersome and not fast enough. An increase in media questions is expected.

b. There is a turnover in personnel and long training periods are required to learn how to properly answer media questions.

c. Answers to media questions must be saved and later searched to avoid different answers to the same question and to reduce duplication of effort.

d. An increase in the amount of documentation is expected.

e. Coordination with the Public Affairs operations within other NASA space centers is anticipated.

To summarize, the aim is to eliminate an employee turnover and training problem for a regularly performed, publicly visible task which is tedious, labor intensive, and can produce an inconsistent result.

2. Progress

Progress to date has been to research hardware and software requirements for a prototype intelligent interactive high speed data search system to automate the manner in which media questions are answered. The effort has led to a prototype design aimed at expository text (technical manuals, reports, regulations, research papers, etc.). The prototype takes advantage of IR techniques to find relevant text, and it uses general world knowledge to answer queries and summarize text. As diagrammed in Figure 1, it is characterized by the following three activities:

I. Automatic indexing based on syntactic, linguistic, and probabilistic techniques to retrieve a ranked list of relevant text for a given natural language query.

II. Converting retrieved relevant text to a conceptual format based on general or world knowledge.

III. Letting meaning be represented by "portions" of the original text and using portions of retrieved text to build a response to a query or generate a summary.

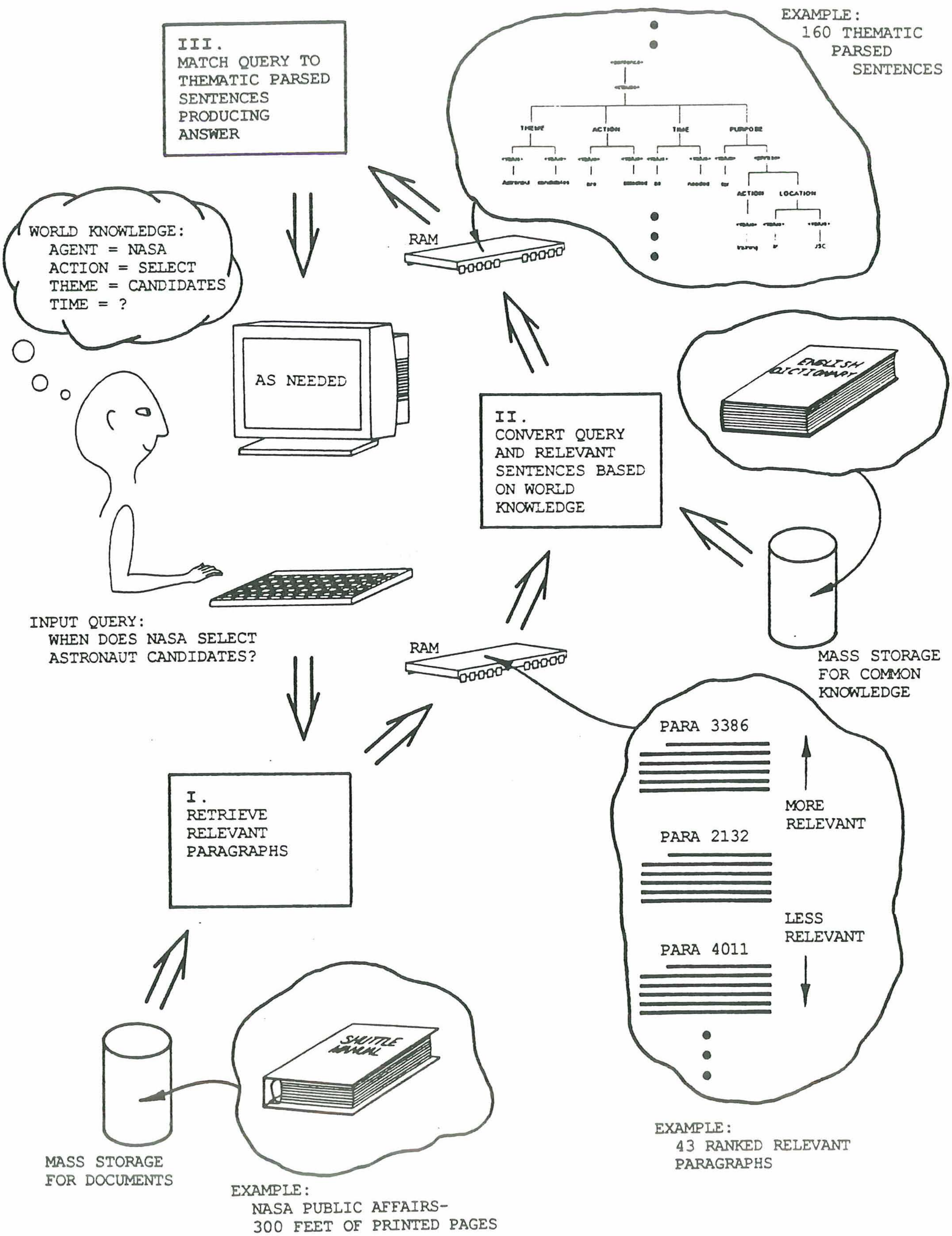


Figure 1: Three Activities for Prototype Question/Answer System.

This design concerns natural language processing and involves the acquisition of general world knowledge. These are complex tasks still open to research and criticism. Consequently, some detail is necessary to make it clear as to what will be accomplished by this project.

3. Method of Attack

Activity I is not new; there are commercial systems which already perform this activity. These systems are complex but rather straightforward to implement. They are based on frequency of occurrence of character strings and the more sophisticated systems know how words should be spelled and what part of speech a word can have (e.g., noun, verb, adjective, etc.). Referring to Figure 1, for a demonstration, the 1,000 page shuttle manual (stored electronically as a three megabyte text file) was used by considering each paragraph of the manual as a document. This resulted in a collection of 4,902 documents. A commercial hypertext system called SPIRIT[3] was used to automatically index the collection and provide natural language access. For the natural language query "When does NASA select astronaut candidates?" a ranked list of forty-three relevant paragraphs was retrieved. Paragraph number 3386 was the most relevant, paragraph number 2132 was the second most relevant, etc. For the prototype system, a ranked list of paragraphs retrieved for a query is stored in RAM to facilitate further processing.

Activity II is based on a linguistic concept called thematic roles. Thematic roles help question answering by revealing how sentence phrases and clauses are related to the verbs and modifiers in a sentence. It is easy to explain thematic roles by using an example and avoiding some detail. Consider the following sentence:

Mary made coffee for John with a percolator.

3 Refer to the DATABASE journal article about SPIRIT in ADDITIONAL SUPPORTING DOCUMENTS.

There are four noun phrases in this sentence, each of which fits into a particular thematic role as follows:

Noun Phrases	Thematic Roles
Mary	**AGENT**
coffee	**OBJECT**
for John	**BENEFICIARY**
with a percolator	**INSTRUMENT**

Four corresponding questions can be answered now:

What was made ? → **OBJECT** → coffee
Who made coffee? → **AGENT** → Mary
For whom was coffee made ? → **BENEFICIARY** → John
With what was coffee made ? → **INSTRUMENT** → a percolator

There are approximately thirty known thematic roles.

It is important to point out that this project is not concerned with figuring out, for example, what "a percolator" is; we only need to know that this string of characters occupies an **INSTRUMENT** position in the thematic form of a sentence. This is perhaps the most important point behind the success of this project.

In order to convert text to thematic form, the following general world knowledge is needed:

a. Certain words suggest specific thematic roles. For example, all verbs other than the "be-verbs" (is, are, was, etc.) become candidates for an **ACTION**. As another example, the word "noon" implies **TIME**.

b. Prepositions and certain conjunctions trigger thematic role possibilities. For example, the preposition "for" can trigger the **BENEFICIARY** or **DURATION** thematic roles. The conjunction "as" can possibly trigger the **MANNER** or **TIME** thematic roles.

c. Normally, verbs cause thematic grids of the form **AGENT** followed by **ACTION** as in

John ran.

or the form **AGENT** followed by **ACTION** followed by **THEME** as in

John chased the dog.

Each of these grids can then be followed by some of the other thematic roles as in

John chased the dog into the house.

Here, "into the house" is a **DESTINATION** thematic role. But there are other thematic grids. For example, there is a class of verbs (represented by the verbs "load" or "spray") which can have a grid which specifies an **AGENT, ACTION, LOCATION, THEME** order, as in

John loaded the truck with furniture.

Here "the truck" is a **LOCATION** rather than a **THEME**.

The conversion process uses the information outlined above for all words which may appear in sentences to be converted. For a somewhat robust system, this would involve approximately 100,000 words. Items B and C are the most important with item C being the most time-consuming to establish. The conversion process can operate even if knowledge for items A and C is not known for domain specific words; this makes the process domain independent.

Acquisition of the above knowledge will be done manually from memory, but also from reading English dictionary entries when we are not so sure. Creating this knowledge is a time consuming process. However, it is felt that the knowledge can be automatically produced directly from standard English dictionary text. We have only just begun to consider procedures for this task. In any case, the knowledge must be established and it can be obtained from an English dictionary. This is shown as an input to Activity II in Figure 1 and labeled as mass storage for common knowledge.

The output of Activity II is a collection of thematically parsed sentences stored in RAM to facilitate further processing. Referring to Figure 1, a thematic parse for the sentence "Astronaut candidates are selected as needed for training at JSC" is shown as an example. This sentence occurred in paragraph number 4011 of the shuttle manual, the third most relevant paragraph for the query "When does NASA select astronaut candidates?" For the forty-three retrieved paragraphs shown as input to Activity II, there are 160 sentences and potentially each one would have to be converted to thematic form.

Activity III is rather straightforward. An answer to the query is generated by a scan of parsed sentences for one having a **TIME** portion, where the **ACTION** specified is "select" and the **THEME** specified is "candidates"; and if an **AGENT** is present, it must have the value "NASA." The response is the **TIME** portion of such a sentence. In Figure 1, this is shown as the phrase "**AS NEEDED**" on the computer monitor, taken from the sentence mentioned above and diagrammed on the RAM device in Figure 1.

A crude version of the SPIRIT system's retrieval capabilities has been implemented in the C programming language for a PC or SUN workstation. We have found that speed is not a problem for this activity; good performance can be achieved in a PC environment. A crude thematic parser is under development now to investigate speed performance in a PC environment for Activities II and III. The Prolog programming language is being used to enable rapid development of the parser. Parallel parsing techniques are also being considered because high speed appears to be an issue here.

The project's three year schedule includes a quickly implemented prototype incorporating all three activities by October 1990. This system will only have a 1000 word vocabulary but should demonstrate relevant text retrieval and thematic parsing ability. Its purpose is to investigate the speed issue mentioned for Activity II. The current level of funding allows for manual development of a 14,000 word vocabulary and demonstration of a question/answer prototype in December 1991, and a summarization prototype in December 1992. These prototypes could handle, for example, the shuttle manual.

4. Relevance to FHTIC Applied Research Grants Objectives

The system called SPIRIT which we used for Activity I in the section Method of Attack is a commercial product available on a PC for $10,000 and on a mainframe computer for $80,000. The system was loaned to the P.I. for demonstration to NASA KSC. Refer to ADDITIONAL SUPPORTING DOCUMENTATION for more information about SPIRIT.

SPIRIT was developed in France. The English version of SPIRIT is not as powerful as the French version. For example, there is no synonym capability in the English version. In addition, the user's manual for the system is in French! It is important to note that we have not been able to find a commercial system in the United States as convenient and powerful as SPIRIT.

The crude version of the SPIRIT system's text retrieval capabilities which we have implemented for the NASA KSC prototype system incorporating all three activities works okay;

but we know now that it should have synonym capability. We feel that addition of the synonym capability and a few other standard information retrieval concepts will make the NASA KSC prototype text retrieval system better than the French commercial product SPIRIT when it comes to English text databases. We also feel that some new thematic research related to that discussed for Activity II and Activity III of the NASA KSC prototype can also be incorporated within the retrieval system to make it truly better than the SPIRIT commercial product.

4.1 First Objective

Our first objective, then, is to carry out this relatively straightforward enhancement and start creating a robust text retrieval system with the intent to make it a commercial product. The planned enhancements, beyond features required for our NASA KSC prototype, can be carried out by one graduate student working half-time for one year and with minimum P.I. supervision. The synonym capability will be incorporated within the NASA KSC prototype before the proposed FHTIC funding period. The graduate student's main activities will be to add thematic techniques, fine tune the system, and create a user's manual. All this is indicated in the Schedule section and reflected in the BUDGET.

We plan to demonstrate the text retrieval system using NASA KSC text data by November, 1991; so several trips to KSC are planned to collect text files. We also plan to demonstrate the system at the Hypertext 91 Conference during November, 1991 using the text from the proceedings of the Hypertext 87, 89, and 91 Conferences.

4.2 Second Objective

The P.I. recently attended the 13th International Conference on Computational Linguistics. This year, the conference technical topics favored automated linguistic analysis using machine readable dictionaries, thesauri, and large common text corpora. It was learned at the conference that much of the text processing research required for completion of Activity II and Activity III

of the NASA KSC funded project is applicable to other endeavors such as machine translation and tools for linguistic analysis. Note that the NASA KSC funded project dictates the development of prototype systems for answering questions and summarizing text.

Consequently, our second objective is to plan for commercialization of the above mentioned text retrieval system and more intelligent text processing applications such as the NASA KSC funded question answer system (prototype available December, 1991), the NASA KSC funded text summarization system (prototype available December, 1992), a grammar correction system, a machine translation system, a toolbox of linguistic analysis programs, etc. The Schedule section and BUDGET reflect 10% release time for the P.I. to develop a comprehensive plan, and outline the expected return. Note that the NASA KSC approved funding as shown on the Schedule for years 1990, 1991, and 1992 represents a significant amount of matching funds for prototype development of these intelligent text processing systems.

5. Schedule

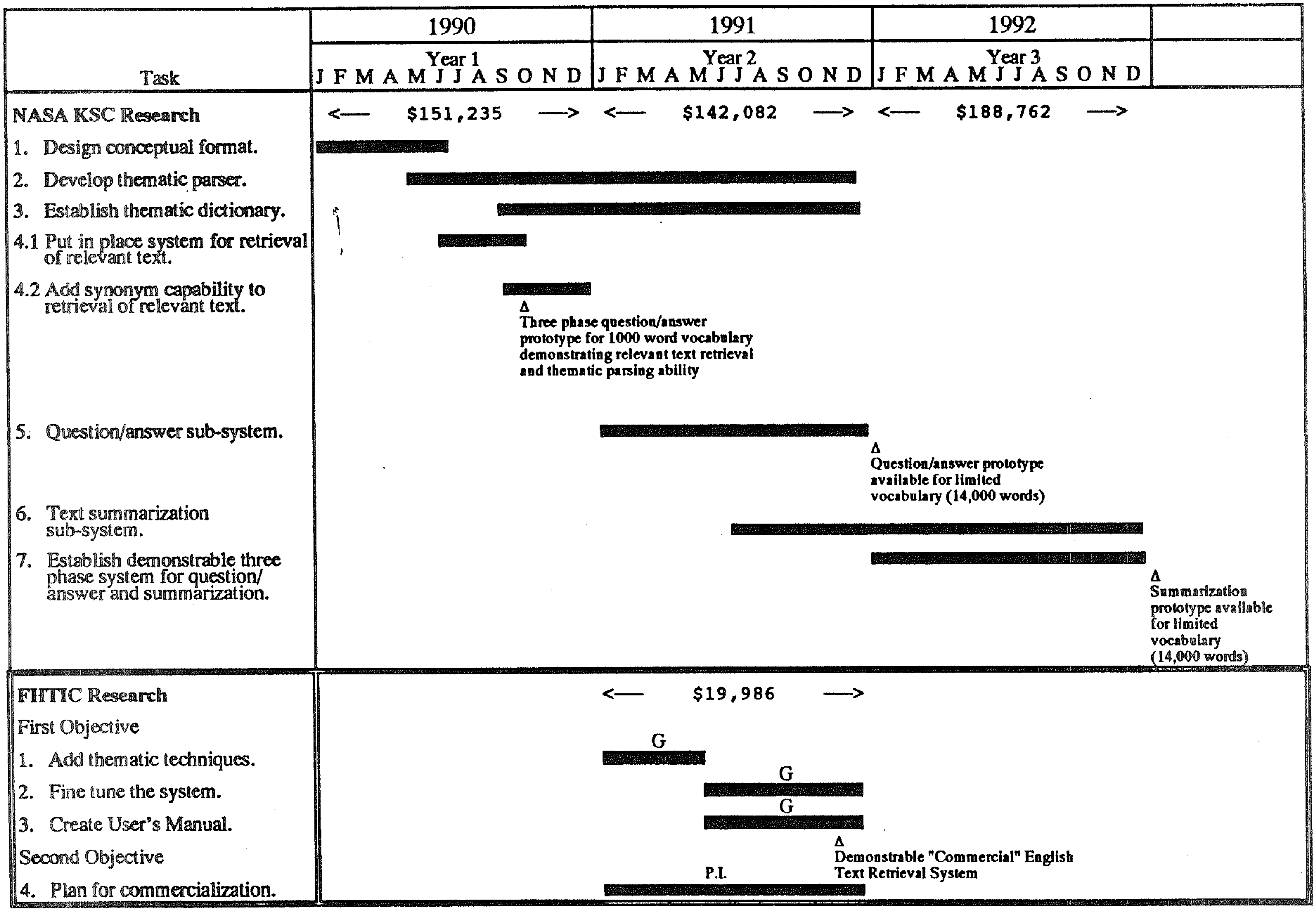
Refer to the chart on the next page for the schedule of activities for this project and its relationship to NASA KSC funded activities.

6. Project Summary

6.1 Technical Approach

The Army Research Institute PM-Trade provided $27,321 for Fall of 1988 to demonstrate the usefulness of thematic roles (surface knowledge) in regard to text processing. NASA KSC provided $33,209 during Summer and Fall of 1989 for the purpose of assessing the situation regarding high speed intelligent text retrieval. A survey of existing technology and commercial systems was performed. The results revealed that Europe is somewhat ahead of the United States in regard to commercialization of text retrieval systems and the application of linguistic research.

SCHEDULE



G = Graduate Student P.I. = Principal Investigator

The currently funded NASA KSC project described in the section Method of Attack makes use of established information retrieval techniques and new linguistic techniques. The linguistic techniques (thematic roles) appear to be consistent with new linguistic strategies and trends recently reported in linguistic journals and especially as reported at the 13th International Conference on Linguistics in Helsinki, Finland, August 15-25, 1990.

Because of all this, we are confident that the technical approach to the NASA KSC project is sound. Please note that NASA KSC has committed close to $500,000 for the demonstration of those existing and new text processing concepts in order to develop a PC-based prototype question answering system and a prototype text summarization system. These systems have application outside the space domain and, indeed, that is one reason why NASA has funded the project. In summary, the P.I. and several graduate students have already put two years of funded research effort into understanding the problem of intelligent text processing and planning for future prototype intelligent text processing systems.

6.2 Qualifications

In the text processing and database technology areas, the P.I. has demonstrated numerous international publications in the past four years, a significant amount of external funding ($627,000), and the ability to develop useful software systems. Furthermore, the P.I. has demonstrated international recognition as being knowledgeable in the area of intelligent text processing; note that he is on the program committee for the RIAO '91 International Conference on Intelligent Text and Image Handling (refer to ADDITIONAL SUPPORTING DOCUMENTATION).

The Department of Computer Science offers three degrees - B.S., M.S., and Ph.D. The department has seven hundred undergraduate majors and 150 graduate students. For 1988-1989, the department obtained external funds in the amount of $2,759,107 which included (1) an award of $1,903.131 which is evenly divided between a Computer Science faculty member, an Electrical Engineering faculty member, and a staff member from the Institute for Simulation and Training,

and (2) awards totalling $966,000 by two Computer Science faculty members working with the College of Engineering. The Computer Science department has access to numerous computing facilities; but note that NASA KSC funds provides all the computer hardware and software needed for the research proposed here.

6.3 Commercialization

As mentioned, most of the proposed effort in this first year of funding is aimed at development of a PC-based text retrieval system similar to a successful French commercial mainframe product which is not easily adaptable for use in the United States. Planning and assessment is proposed for more intelligent text processing software. After this first year of funding, the PC-based text retrieval system will exist and be demonstratable; only its promotion and marketing as a commercial product will remain. This may occur via an industrial partner or through UCF small business or marketing channels. A similar scenario is expected for the more intelligent text processing software such as the question answering system or text summarization system for which NASA KSC funds are being used to develop prototypes.

6.4 External Support for the Project

External support for this project already exists and is detailed in section 6.1 above. Via the NASA KSC funded research, various NASA subcontractors have heard about our research efforts and have expressed an interest in using the prototypes we are developing.

6.5 Florida University Infrastructure

At this point in time, no cooperative activities with other Florida universities exist. But an attempt will be made during the planning phase to identify possible cooperative activities. In regard to development of students for Florida industry, one particular NASA KSC subcontractor has been attending our project presentations and has requested the P.I. to identify students seeking employment who have experience related to the project.

FLORIDA HIGH TECH COUNCIL
APPLIED RESEARCH GRANTS PROGRAM

BUDGET FORM

Principal Investigator/Program Director: Dr. James R. Driscoll

DETAILED BUDGET FOR 12 MONTH PERIOD				From Jan. 1, 1991		Through Dec. 31, 1991
Personnel		**Time/Effort**		**Dollar Amount Requested**		
Name	Postition/Title	%	Hours per week	Salary	Fringe Benefits	Totals
Dr. James R. Driscoll	P.I.	10	4	6,236.00	1,692.00	7,928.00
TBA	Graduate Student #1	50	20	11,440.00	18.00	11,458.00
			Subtotals	17,676.00	1,710.00	19,386.00

Equipment	
Supplies Computer disks, paper and copies	100.00
Other Expenses	
Travel Local travel to KSC (25 trips @ 100 mi. @ $.20/mi)	500.00
Consultant Costs	
Other Miscellaneous Costs	
Total Costs	$19,986.00

PRINCIPAL INVESTIGATOR/PROGRAM DIRECTOR: Dr. James R. Driscoll

Budget for Entire Project Period

In order to gain a funding history for this grant, indicate actual budget for each year funded and/or proposed. Also check if "Funded" or "Proposed" for each year (ie, new proposals will be "Proposed" for all years).

Budget CategoryTotals	1st Budget Period	Additional Years of Support			
		2nd	3rd	4th	5th
Personnel (Salary and fringe benefits)	$ 19,386.00				
Equipment					
Supplies	$ 100.00				
Other Expenses					
Travel	$ 500.00				
Consultant Cost					
Other Miscellaneous Costs					
Total Costs	$ 19,986.00				
Budget Status (✓)	___ Funded ___ Proposed	___ Funded ___ Proposed	___ Funded ___ Proposed	___ Funded ___ Proposed	___ Funded ___ Proposed
Total Costs For Entire Proposed Project Period				$ 19,986.00	

Budget Justification:

1. The salary for Dr. Driscoll reflects (1) extra supervisory activity and (2) extra fact finding activity beyond that required for the existing NASA KSC project.
2. The salary for graduate student #1 is for implementation effort beyond that required under the existing NASA KSC project.
3. All computer hardware and software is available via the existing NASA KSC funded project.
4. Supply expenses are for those beyond the existing NASA KSC project.
5. Travel to conferences and other expenses is already provided by the existing NASA KSC project.
6. Approximately twenty-five trips to KSC beyond the existing NASA KSC project requirements are expected.

PERSONNEL

Name:	James R. Driscoll
Title:	Associate Professor of Computer Science
Birth Date:	January 18, 1948
Address:	Department of Computer Science University of Central Florida P. O. Box 25000 Orlando, Florida 32816
Home:	8380 Roanne Drive Orlando, Florida 32817
Telephone:	(407) 275-2341 (office) (407) 677-1108 (home)

EDUCATION

University of Kansas	B.S.	1971	Electrical Engineering (with Highest Distinction)
University of Kansas	M.S.	1974	Computer Science (with honors)
University of Kansas	Ph.D.	1977	Computer Science

M.S. Thesis: "Design of a Run-Time System for ALGOL 60"
D. Dissertation: "Towards the Design and Implementation of a Unified Data Based Management System"

PROFESSIONAL EXPERIENCE

Associate Professor, Department of Computer Science, University of Central Florida, 1981 to present.

Assistant Professor, Department of Computer Science, University of Central Florida, 1976-1981.

Instructor, Department of Computer Science, University of Kansas, 1974-76.

Graduate Teaching Assistant, Department of Computer Science, University of Kansas, 1972-74.

PROFESSIONAL DEVELOPMENT

Data Base Advisor, Institute for Simulation and Training, University of Central Florida, May 1986 to May 1988. Artificially Intelligent Keywording System.

Consultant, Insystec Corporation, Orlando, Florida, 1981-82. Data Base Management Systems.

Consultant, Harris Corporation Controls Group, Melbourne, Florida, 1977-78. Data Base Control of Power Systems.

RESEARCH IN PROGRESS

Development of a domain independent natural language interface to document databases, a domain independent diagnostic system, and domain independent systems for document abstracting, summarization, merging, language translation, and grammar correction.

RELEVANT PUBLICATIONS

L. C. Malone, J. R. Driscoll and J. W. Pepe, "Modeling the Performance of an Automated Keywording System" to appear in the journal Information Processing and Management.

R. Driscoll, D. A. Rajala, W. H. Shaffer and D. W. Thomas, "The Operation and Performance of an Artificially Intelligent eywording System," to appear in the journal Information Processing and Management.

C. Malone, J. W. Pepe and J. R. Driscoll, "Evaluation of an Automated Keywording System" Microcomputers for Information gement, June 1990.

K. Ray and J. R. Driscoll, "New Directions for Microcomputer-based Hypertext Systems" DATABASE Magazine, June, 1990.

L. Bucher and J. R. Driscoll, "Automated Abstracting Using Thematic Roles," Proc. of the Third Pan Pacific Computer Conference, Beijing, China, August 15-19, 1989.

I. Syu and J. R. Driscoll, "Portability in Natural Language Query Interfaces via Thematic Roles," Proc. of the Third Pan Pacific Computer Conference, Beijing, China, August 15-19, 1989.

C. E. Couden and J. R. Driscoll, "Generating Responses with Constructive Contributions for Queries of Expository Text Databases," Proc. of the Third Pan Pacific Computer Conference, Beijing, China, August 15-19, 1989.

S. D. Lang, J. R. Driscoll and J. H. Jou, "A Unified Analysis of Batched Searching for Sequential and Tree Structured Files," Transactions on Data Base Systems (TODS), Vol. 14, NO. 4, December 1989.

J. R. Driscoll, W. Lee, I. Syu, T. G. Rackley and R. D. Capetillo, "Easing the Task of Referencing Electronically Stored Troubleshooting Manuals," Proc. of the Conference of the International Association of Knowledge Engineers (IAKE 89), College Park, Maryland, June 26-28, 1989.

L. Bucher and J. R. Driscoll, "An AI Approach to Automated Abstracting," Proc. of the Fourth Rocky Mountain Conference on Artificial Intelligence, Denver, Colorado, June 8-9, 1989.

J. W. Smith and J. R. Driscoll, "An Artificially Intelligent Text Summarization Algorithm," Proc. of the Fourth Rocky Mountain Conference on Artificial Intelligence, Denver, Colorado, June 8-9, 1989.

I. Syu and J. R. Driscoll, "A Portable Natural Language Query Interface," Proc. of AVIGNON 89: Ninth International Workshop on Expert Systems and their Applications," Avignon, France, May 29-June 2, 1989.

J. R. Driscoll, D. A. Rajala, W. H. Shaffer and D. W. Thomas, "An Application of Artificial Intelligence Techniques to Automated Keywording," Proc. of RIAO88 Conference on User-Oriented Content-Based Text and Image Handling, MIT, Cambridge, Massachusetts, March 1988.

J. R. Driscoll, S. D. Lang and L. A. Franklin, "Modeling B-tree Insertion Activity," Information Processing Letters, Vol. 26, No. 1, September 1987.

J. R. Driscoll, S. D. Lang and S. M. Bratman, "Achieving Minimum Height for Block Split Tree Structured Files," Information Systems (GB), Vol. 12, No. 1, 1987.

J. R. Driscoll, H. N. Srinidhi and T. S. Chesser, "A Network Technique to Achieve Program and Data Security with Nominal Communications Overhead," Proc. of the 1986 ACM/IEEE Fall Joint Computer Conference, Dallas, TX, November 1986.

S. D. Lang, J. R. Driscoll and J. H. Jou, "Batch Insertion for Tree Structured File Organizations--Improving Differential Database Representation," Information Systems (GB), Vol. 11, No. 2, 1986.

S. D. Lang, J. R. Driscoll and J. H. Jou, "Improving the Differential File Technique via Batch Operations for Tree Structured File Organizations," Proc. of the Second Annual IEEE Data Engineering Conf., Los Angeles, CA, February 1986.

K. Kinsley and J. R. Driscoll, "A Generalized Method for Maintaining Views," Proc. of the 1984 National Computer Conference (AFIPS press), Las Vegas, Nevada, June 1984.

S. Masters and J. R. Driscoll, "An Evaluation of RQL: A Relational Data Base System for a Low-End Microcomputer Configuration," ACM SIGSMALL Newsletter, May 1983.

S. Masters and J. R. Driscoll, "RQL: A Relational Data Base System for a Low-End Microcomputer Configuration," Proc. of Workshop on Relational DBMS Design/Implementation/Use on Microcomputers, Toulouse, France, February 1983.

K. S. Barley and J. R. Driscoll, "A Survey of Data Base Management Systems for Microcomputers," BYTE Magazine, November 1981.

R. C. Brigham, R. D. Dutton and J. R. Driscoll, "Complexity of a Proposed Data Base Storage Structure," Information Systems (GB), Vol. 6, No. 1, 1981.

V. Gharavi and J. R. Driscoll, "The Implementation of Dynamic Derived Relations," Proc. of the COMPSAC Fourth International Conference, Chicago, Illinois, October 1980.

C. H. Chen, J. R. Driscoll and K. Grammel, "Physical Storage and Physical Navigation Within the Relational DBMS RAQUEL II," Proc. of the Eighth Texas Conference on Computing Systems, Dallas, Texas, November 1979.

. A. Dutton, C. H. Chen and J. R. Driscoll, "A Relational DBMS Conforming to an Architecture Which Incorporates a Physical Storage Language and a Physical Navigation Language," Proc. of the COMPSAC Third International Conference, Chicago, Illinois, November 1979.

`. Kinsley and J. R. Driscoll, "Dynamic Derived Relations Within the RAQUEL II DBMS," Proc. of the ACM 79 National Conference, oit, Michigan, October 1979.

T. Bonar and J. R. Driscoll, "A Very Easy Hierarchical DBMS Implementation," Proc. of the ACM 79 National Conference, Detroit, Michigan, October 1979.

J. R. Driscoll, B. A. Dutton and K. C. Kinsley, "A Relational Storage Scheme Suitable for Derived Views," Proc. of the ACM 78 National Conference, Washington, D.C., December 1978.

J. R. Driscoll, "Two Languages, and their Environment, Founded on Basic DBMS Constructs," Proc. of the Third Jerusalem Conference on Information Technology, Jerusalem, August 1978.

J. R. Driscoll and Y. E. Lien, "A Selective Traversal Algorithm for Binary Search Trees, "Communications of the ACM, U.S.A., June 1978.

J. R. Driscoll, "The Physical Representation of Data Models," Proc. of ICMOD 78, Milan, Italy, June 1978.

Y. E. Lien, C. E. Taylor, J. R. Driscoll and M. L. Reynolds, "Binary Search Tree Complex--Towards the Implementation of Relations," Proc. Int. Conf. on Very Large Data Bases, Framingham, Massachusetts, September 1975.

GRANTS

NASA Kennedy Space Center, "Intelligent Interactive High Speed Data Search System (Year 2, Year 3)," J. Driscoll, Summer Semester 1990 through Fall Semester 1992, $330,000 (approved but award pending).

NASA Kennedy Space Center, "Intelligent Interactive High Speed Data Search System (Phase II)," J. Driscoll, Summer and Fall Semester 1990, $68,361.00 (awarded June 1990).

NASA Kennedy Space Center, "Intelligent Interactive High Speed Data Search System (Phase I)," J. Driscoll, Spring Semester 1990, $82,874 (awarded December 1989).

NASA Kennedy Space Center, "Public Affairs Q&A System," J. Driscoll, Summer Semester 1989, $33,209 (awarded May 1989).

DoD Army Research Institute/PM-Trade, "Conceptual Information Extraction and Summarization for Expository Text," J. Driscoll, ll Semester 1988, $27,321 (awarded August 1988).

Institute for Simulation and Training (IST), "The Application for Statistical Techniques to Improve the JULLS Automated Keywording System," L. Malone and J. Driscoll, Fall Semester 1987, $9,951 (awarded August 1987).

Institute for Simulation and Training (IST), DoD Training and Performance Data Center (TPDC), "Data Base and AI System Technology," J. Driscoll - Salary support for Fall Semester 1987 and Spring Semester 1988, $25,142 (awarded August 1987).

Institute for Simulation and Training (IST), DoD Training and Performance Data Center (TPDC), "Automated Keywording," J. Driscoll - Salary support for Summer Semester 1987, $15,963 (awarded May 1987).

Institute for Simulation and Training (IST), DoD Training and Performance Data Center (TPDC), "Automated Keywording, Collective Training Report, and Functional Description," J. Driscoll - Salary support and publication expenses for Spring Semester 1987, $13,112 (awarded December 1986).

Institute for Simulation and Training (IST), DoD Training and Performance Data Center (TPDC), "Joint Universal Lessons Learned System," J. Driscoll - salary support for Summer Semester 1986, $20,930 (awarded May 1986).

University of Central Florida In-House Grant, "Computer Program and Data Security," J. Driscoll - Salary support for Summer Semester 1985, $5,112 (awarded March 1985).

INTEL Computer Corporation, "Minimization of Logic and Data Traffic in Distributed Networks," J. Driscoll and H. Srinidhi - computer equipment and software, $40,000 (awarded December 1984).

Contract, "INSYSTEC Corporation and the University of Central Florida Purchase Agreement," J. Driscoll - salary support on an hourly basis for "on-campus" and "off-campus" work (awarded November 1981).

NSF LOCI Grant, "Total Microprocessor System Design and Development," R. Guha and J. Driscoll - salary support until September 1983, $46,248 (awarded September 1981).

University of Central Florida In-House Grant, "An Evaluation of Two Methods for Implementing Dynamic Derived Relations," J. Driscoll - salary support for Summer Quarter 1980, $2,340 (awarded January 1980).

University of Central Florida In-House Grant, "A Rational Data Base Management System Supporting Dynamic Derived Relations," J. riscoll - salary support for Summer Quarter 1979, $3,008 (awarded November 1978).

University of Central Florida In-House Grant, "A Data Base Management Systems Supporting Multiple User Views of a Single Data Base," J. Driscoll - salary support for Summer Quarter 1978, $2,780 (awarded November 1977).

CURRENT INTERESTS AND RESEARCH

Data Base Management Systems
Systems Design
Analysis of Algorithms and Data Structures
AI Based Intelligent Data Base Systems
Expert Systems
Hypertext and Hypermedia

DATA BASE SYSTEMS EXPERIENCE

dBASE III Plus
Clipper
FOCUS
IMS
Oracle (on Harris Computer)

ACHIEVEMENTS

Developed a novel Sprinkling System Controller using microprocessor technology. A patent is being applied for.

Developed a relational DBMS for the Apple II microcomputer which was sold nationally as an educational tool via HELLO Software.

Developed several data base systems for the IBM PC which were also distributed to universities for use in data base courses.

NUMBER OF DOCTORAL STUDENTS DIRECTED: 1

NUMBER OF MASTER'S STUDENTS DIRECTED: 20

COMMUNITY SERVICE

Faculty Advisor for the University of Central Florida ACM Student Chapter (September 1976 through June 1980)

Acting Chairman for the Central Florida Chapter of the ACM (October 1978 through April 1980)

RECOGNITIONS

1971	The award of Highest Distinction in the completion of the B.S. Degree by the University of Kansas.
1974	The award of Honors in completion of the requirements of the M.S. Degree by the Department of Computer Science at the University of Kansas.
1979	Listed in Who's Who in Technology.
1982	Awarded second place for the article "A Survey of DBMSs for Microcomputers" in the reader's poll of the November 1981 issue of BYTE Magazine.

PROFESSIONAL ORGANIZATIONS

Association of Computing Machinery (ACM)
IEEE Computer Society
International Association of Knowledge Engineers (IAKE)

OTHER SUPPORT RECEIVED OR PENDING

NASA Kennedy Space Center, "Intelligent Interactive High Speed Data Search System (Year 2, Year 3)," J. Driscoll, Summer Semester 1990 through Fall Semester 1992, $330,000 (approved but award pending).

NASA Kennedy Space Center, "Intelligent Interactive High Speed Data Search System (Phase II)," J. Driscoll, Summer and Fall Semester 1990, $68,361 (awarded June 1990).

EXTERNAL SUPPORT, INDUSTRIAL/FEDERAL COLLABORATION

NASA Kennedy Space Center, "Intelligent Interactive High Speed Data Search System (Year 2, Year 3)," J. Driscoll, Summer Semester 1990 through Fall Semester 1992, $330,000 (approved but award pending).

NASA Kennedy Space Center, "Intelligent Interactive High Speed Data Search System (Phase II)," J. Driscoll, Summer and Fall Semester 1990, $68,361 (awarded June 1990).

NASA Kennedy Space Center, "Intelligent Interactive High Speed Data Search System (Phase I)," J> Driscoll, Spring Semester 1990, $82,874 (awarded December 1989).

NASA Kennedy Space Center, "Public Affairs Q&A System," J. Driscoll, Summer Semester 1989, $33,209 (awarded May 1989).

DoD Army Research Institute/PM-Trade, "Conceptual Information Extraction and Summarization for Expository Text," J. Driscoll, Fall Semester 1988, $27,321 (awarded August 1988).

OTHER SUS INTERACTIONS

No collaboration exists at the present time, but during the proposed planning phase, an effort will be made to identify possible interactions.

TECHNOLOGY TRANSFER

Beyond the usual techniques of publication in conference proceedings and journals, the principal investigator is planning an intelligent text retrieval product demonstration in November 1991 at the Hypertext 91 Conference.

ADDITIONAL SUPPORTING DOCUMENTATION

1. **RIAO 91 Intelligent Text and Image Handling - Call for Papers**

2. **RIAO 91 Intelligent Text and Image Handling - Call for Product Demonstrations**

3. **DATABASE Journal Article about SPIRIT**

stamp

RIAO 91
Center for the Advanced Study of Information
Systems, Inc. (CASIS)
Ms. M.-T. MAURICE
220 East 72nd Street #10F
New York, N.Y. 10021
U.S.A.

Paper submitted (4 copies, 20 pages maximum) must arrive before October 30, 1990.

* CID :
36 bis rue Ballu F-75009 PARIS FRANCE
Tel. (33) (1) 42 85 64 75 Fax : (33) (1) 45 26 64 45

* North America
Ms M.-T. MAURICE
220 East 72nd Street #10F New York NY 10021 USA
Tel (212) 879 48 19 Fax (212) 685 81 86

Each oral presentation will last 20 minutes followed by 10 minutes of discussion.

Certain papers will be published in a more detailed presentation in the journal: "INFORMATION PROCESS AND MANAGEMENT"

English will be the working language of the conference.

PROGRAM COMMITTEE

Chairman

J. ARSAC

Professor of University of Paris VI

J. ANDRE (F) IRISA Rennes
J.C. BASSANO (F) Univ. d'Orléans
E. BAUER-BERNET (B) Univ. Libre de Bruxelles
A. BOOKSTEIN (USA) Univ. of Chicago
P. BRUNET (SP) Univ. Polytécnica Catalunya
C. CHEN (USA) Simmons College
C. CHRISMENT (F) Univ. de Toulouse
M. CREHANGE (F) Univ. de Nancy
M. DILLON (USA) OCLC
J. DRISCOLL (USA) Univ. of Central Florida
C. FLUHR (F) Univ. ParisXI/INSTN
E. FOX (USA) Virginia Polytech. Inst.
C. FRASSON (CAN) Univ. de Montréal
R. GALLAIRE (F) Bull SA Paris
A. GAMACHE (CAN) Univ. Laval
E. GARCIA-CAMARERO (SP) Univ. Complutense Madrid
P. GOOSSENS (B) Bibliothèque Royale
G. GREFENSTETTE (F) Univ. Toulouse
[illegible] (S) CBS DOXUM
C. JONCKHEERE (NDL) Inst. European Bureau
R. KUHLEN (FRG) Univ. de Constance
A. LELU (F) CNRS/INIST Nancy
B. LUCARELLA (I) Univ. Milano
J. MULLENDERS (B) Alcatel/Bell
S. OHSUGA (JP) Univ. of Tokyo
A.P. PARKES (GB) Univ. of Lancaster
A. [illegible] (F) LIMSI/Paris XI
L. RAU (USA) General Electric
J. ROUAULT (F) Univ. de Grenoble
T. SARACEVIC (USA) Rutgers Univ.
C. SCHWARZ (FRG) Siemens
A. TURNER (F) CNRS/DIST Nancy
B. VICKERY (GB) Tom Am. Inc.
J.J. VILLANUEVA (SP) U.A.B.
D. WALKER (USA) Bell Comm. Res.
A. ZAMPOLLI (I) Univ. of Pisa

Coordinator in Spain:
C. CARACUBERTA U.A.B.

R.I.A.O 91 — CALL FOR PAPERS

INTELLIGENT TEXT AND IMAGE HANDLING

Universitat Autònoma de Barcelona
Barcelona, Spain - April 2-5, 1991
Sponsored by the

European Economic Community
and
Minister of "Industrie et Aménagement du Territoire", France
Minister of "Recherche et Technologie, France
President of the "Generalitat de Catalunya", Spain
Rector of the "Universitat Autònoma de Barcelona", Spain

Organized by the:

CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE (C.I.D.)
&
CENTER FOR ADVANCED STUDY OF INFORMATION SYSTEMS, Inc. (CASIS)

with the participation of the
Centre National de la Recherche Scientifique (CNRS)
France Telecom
Institut National de Recherche en Informatique et Automatique (INRIA)
and the
Federación Española de Sociedades de Archivística, Biblioteconomía y Documentación (FESABID)
Instituto de Información y Documentación en Ciencia y Tecnología (ICYT)
American Federation of Information Processing Societies (AFIPS)
This conference is prepared under the direction of
Professor André LICHNEROWICZ
of the Academy of Sciences of Paris

RIAO - Recherche d'Information assistée par ordinateur
(Computer-aided information research)

GENERAL INTRODUCTION

The purpose of this Conference is to present the state of the art in the storage, retrieval and diffusion of non-structured information found in text, image and sound.

This field is developing rapidly : the entire information technology industry is tending towards the convergence of computing, telecommunications and audiovisual techniques. Conditions continually improve for satisfying users' needs and desires for extensive and convivial access to information.

The previous state of the art in this field, "RIAO 88" held at M.I.T. (Cambridge - U.S.A.) in March 1988, was a resounding success, bringing together members of the international scientific community working in these fields.

"RIAO 91" will take place at the Universitat Autònoma de Barcelona (Catalonia, Spain) from April 2-5, 1991. This conference will present, on one hand, recent scientific research, and on the other, demonstrations of prototypes resulting from this research as well as the most innovative new products appearing on the market.

This call for papers is addressed to researchers from all countries, engaged in academic or industrial research.

CALL FOR PAPERS

GENERAL THEME :

Full-text and heterogeneous media data bases are characterized by the fact that the structure of the information that they contain can rarely be known a priori. Traditional hierarchical and relational database management systems provide inadequate treatment. The absence of homogeneous structure and the great diversity of information in even moderately sized bases make it difficult to foresee which sets of questions will be asked. Information research remains a hard problem, yet computing techniques and technologies seem to provide more power than is being used.

You are invited to submit papers showing how these problems of storage, research and diffusion of non-structured information can be solved.

Particular attention will be given to the following themes:

- techniques for reducing imprecision in localizing information in full text
- data input control and verification
- end user interfaces
- new media

A large number of specific subjects can be treated within this general theme.

SPECIFIC THEMES

A - Linguistic analysis for automatic text treatment
  A1. Automatic indexation
  A2. Automatic abstracts
  A3. Natural language interrogation
  A4. Multilingual interfaces
B - Construction and utilization of large linguistic knowledge bases (electronic dictionaries, thesaurus, bilingual dictionaries)
C - Confidentiality in information retrieval systems
D - Multilingual interrogation and computer assisted translation
E - Automatic extraction of factual information from full text
F - User interfaces and ergonomy of information research systems
G - Artificial intelligence for user aid and for personalizing systems
H - Intelligent navigational aids and automatic data structuring in hypertexts and hypermedia
I - Neural nets for computer aided information research
J - Data entry systems (OCR, automatic structure recognition, document preparation standards...)
K - New applications:
  K1. Software engineering and information research systems, program and documentation retrieval for re-use, production of intelligent tutorial and documentation systems in software development.
  K2. Automatic image indexation via pattern recognition
  K3. Optical memories (videodisk, CD-ROM, CD-I, optical numerical disks)
  K4. Multimedia systems managing text, sound and images
  K5. Content addressable electronic mail systems
  K6. Voice entry and speech recognition

CONDITIONS

In order to be accepted, the papers must be validated by a prototype or a working model. The authors may be asked to demonstrate their prototype or working models to a member of the program committee. We expect the authors to give a demonstration of their system during the conference, at a time separate from their oral presentation. It is advisable that these demonstrations run on standard material.

APPLICATION FORM

NAME : ______ First ______ Last
TITLE/POSITION :
ORGANIZATION:
ADDRESS:
CITY: ______ STATE: ______ ZIP :
COUNTRY : ______ FAX : ______ Electronic mail :
TELEPHONE:
I plan to attend the conference, please send me the program: YES NO
I plan to present a paper: YES NO
Conference theme (circle one): A B C D E F G H I J K
Title of the communication:
Are you willing to give a demonstration of your prototype? YES NO
Equipment needed:
Please mail this form before October 30, 1990

SELECTION CONDITIONS

- Please, return the attached "Application Form" before September 30, 1990 with trade documentation concerning the product to be demonstrated.

- A questionnaire will be sent back to you. This must be returned by October 30, 1990.

- A preliminary selection will be made by the committee of experts based on this information, and the selected applications will then be submitted to a more detailed expertise.

- The final decision of the committee will be released on December 20, 1990.

For more information, please contact:
CID :
34 bis rue Balla
F-75009 PARIS
FRANCE
Tel.: (33) (1) 42 85 04 75
Fax : (33) (1) 46 26 84 45

RIAO 91

CALL FOR PRODUCT DEMONSTRATIONS

INTELLIGENT

TEXT AND IMAGE HANDLING

Universitat Autònoma de Barcelona
Barcelona, Spain - April 2-5, 1991

Sponsored by the

European Economic Community
and the
Minister of "Industrie et Aménagement du Territoire", France
Minister of "Recherche et Technologie", France
President of the "Generalitat de Catalunya", Spain
Rector of the "Universitat Autònoma de Barcelona", Spain

Organized by the:

CENTRE DE HAUTES ETUDES INTERNATIONALES
D'INFORMATIQUE DOCUMENTAIRE
(C.I.D.)

with the participation of the
Centre National de la Recherche Scientifique (CNRS)
France Telecom
Institut National de Recherche en Informatique et Automatique
(INRIA)
and the
Federación Española de Sociedades de Archivística, Biblioteconomía
y Documentación (FESABID)
Instituto de Información y Documentación en Ciencia y Tecnología
(ICYT)
American Federation of Information Processing Societies (AFIPS)

This conference is prepared under the direction of
Professor André LICHNEROWICZ
of the Academy of Sciences of Paris

RIAO = Recherche d'information assistée par ordinateur
(Computer-aided information retrieval)

GENERAL INTRODUCTION

The purpose of this Conference is to present the state of the art in the storage, retrieval and diffusion of non-structured information found in text, image and sound.

This field is developing rapidly : the entire information technology industry is tending towards the convergence of computing, telecommunications and audiovisual techniques. Conditions continually improve for satisfying users' needs and desires for extensive and convivial access to information.

The previous state of the art in this field, "RIAO 88" held at M.I.T. (Cambridge - U.S.A.) in March 1988, was a resounding success, bringing together members of the international scientific community working in these fields.

"RIAO 91" will take place at the Universitat Autònoma de Barcelona (Catalonia, Spain) from April 2-5, 1991. This conference will present, on one hand, recent scientific research, and on the other, demonstrations of prototypes resulting from this research, as well as the most innovative new products appearing on the market.

The call for papers is addressed to researchers from all countries, engaged in academic or industrial research.

This call for product demonstrations is addressed to European companies or organizations, marketing hardware or software related to the conference themes.

CALL FOR PRODUCT DEMONSTRATIONS

Companies and organizations which market innovative and competitive hardware or software related to the conference themes should exhibit at RIAO 91 for these reasons:

- The products selected for demonstration (no more than thirty) are chosen by a committee of experts on the basis of *innovation, and responsiveness to current and future market needs.* This selection is itself a guarantee of quality which can be used to the product's advantage.

- A free space will be reserved for the demonstration of the industrial products selected.

- RIAO 88 has shown that this exhibition has been very beneficial, as previously selected companies will testify.

- At RIAO 91, the marketing agent will be able to contact potential clients directly, gauge the competition, and meet the best specialists in the domain who may be able to help in future development.

- The participants will benefit from the extensive communication effort associated with the exhibition: a catalog , a videocassette, numerous articles in the international press, as well as a deal-making session organized during the conference to bring together marketing, development and research partners.

GENERAL THEMES

The exhibition will highlight hardware and software that facilitate the storage , management , retrieval and diffusion of non-structured information present in text, images, or sound. A list of specific themes follows:

SPECIFIC THEMES

A - *Text retrieval systems incorporating:*
- A1 . Automatic indexation using linguistic tools
- A2 . Automatic abstracting
- A3 . Natural language interrogation
- A4 . DBMS incorporating full text
- A5 . Multilingual interfaces
- A6 . Speech interfaces
- A7 . Real-time updating
- A8 . Hypertext, hypermedia
- A9 . Query reformulation
- A10. Inter-system gateways

B - Information entry systems: B1 . *Optical character recognition*
- B2 . Standards (SGML, ODA, ...)
- B3 . Sound, image, and text compression

C - Archiving systems (CD-ROM, CD-V, CD-I, CD-XA...)

D - Help systems
- D1. Lexical data bases
- D2. Conceptual graphs

E - New visualization technologies (HDTV, imagers...)

F - User environment:
- F1. Specialized architectures
- F2. Networks
- F3. Workstations
- F4. Electronic mail
- F5. Computer assisted translation

OTHER DATA REQUIREMENTS

1. **Personnel**

2. **External Support**

Page intentionally left blank

Sate of Florida
High Technology and Industry Council
Applied Research Grants Program

Faculty Profile Form

Personal Information

Name Driscoll
(Last)

Please Print

James R.
(First) *(Middle Initial)*

Title Associate Professor

Dept/Institute/Center Dept. of Computer Science

College College of Arts & Sciences

University University of Central Florida

Expertise Narrative

Please describe in detail your overall areas of expertise as they pertain to this project

In the last four years, Dr. Driscoll has produced eighteen international conference and journal publications in the areas of efficient file organization, and automatic text processing (automatic keywording, text retrieval, and intelligent text manipulation). For the past three years, he has been continuously funded in these areas by DoD Training Performance Data Center (TPDC), Army Research Institute (ARI)/PM-Trade, and NASA Kennedy Space Center. Total funding from these sources has been $627,000. The work funded by DoD TPDC led to the development of a computer system which mimicked the behavior of an expert human keyworder and assigned keywords to military lessons learned from war games, excercises, and real military confrontations. The system is in use today. The work funded by ARI/PM-Trade involved a demonstration that general world knowledge (surface knowledge) could be used to automate the activity of searching military maintenance manuals in order to match problem descriptions to symptoms in search of corrective procedures. The work funded by NASA KSC concerns efficient retrieval of text information. Dr. Driscoll is a member of the program committee for RIAO '90 International Conference on Intelligent Text and Image Handling.

Sate of Florida
High Technology and Industry Council
Applied Research Grants Program

Industry Profile Form

Company Name

Please Print

John F. Kennedy Space Center, NASA

Address

Kennedy Space Center (City)

Florida (State)

32899 (Zip)

Expertise Narrative

Please describe in detail your company's research and development activities

The Advanced Operations Program within NASA is a technology demonstration program which seeks, through its projects, to enhance shuttle operations by the reduction of costs, the increase of efficiency, and the improvement of safety. A majority of the funds expended by the program are to support software projects because it is through the application of artificial intelligence, expert systems, data-base systems, and other newly developed software techniques that the most significant near-term benefits to shuttle operations are foreseen. At KSC, the projects include Remote Maintenance Monitoring System, Operations Analyst for a Distributed System, Intelligent Launch Decision Support System, Main Propulsion System Pneumatic, Intelligent Computer Aided Trainer, Computer CARE Center, Natural Language Knowledge Acquisition System, Ground Operating Simulation Technique, Intelligent Interactive High Speed Data Search System, and Intelligent Interactive Visual DBMS.

Point of Contact

For additional information concerning this company/business, please contact:

Name **Please Print**

Davis (Last)

Tom (First)

(Middle Initial)

Title NASA Technical Officer for Cooperative Agreement NCC10-0003 S-2

Telephone (407) 867-2780 (Area Code)

Section 5.

An Analysis of Natural Language Questions

Index:

1. Non-question information

A. Thematic Roles and their definitions(including alternate names.

1) Verbs

a) ACTION denotes movement or activity that can be seen or heard. ex: Ice drips. ACTION can also be designated affective or causative.

b) PROCESS refers to internal activities (i.e seeing, hearing) or to changes in condition of persons or things that are experiencers or patients. ex: Ice melts. PROCESS can also be designated affective or causative.

c) STATIVE denoting persons or things that are in a particular state or condition. Verbs in Patterns 3, 4, and 5 below are stative most of the time. "However, verbs such as STAY and REMAIN, which may be stative verbs, are commonly intransitive." STATIVE can also be designated ambient, static, or dynamic.

2) Other Words

a) MOVER designates a person or animate creature that performs an action.

b) PATIENT a person or object on whom an action is performed or who receives the effect of an action or process.

c) EXPERIENCER a person or animate being that undergoes or experiences a change.

d) INSTRUMENT subject of an action or process verb OR an object that must be held and manipulated by some person. Something that has a part in bringing about an action or process but is not the instigator. (Can this also include manipulation by a machine?)

e) ENTITY person or thing in a particular state or condition (subject of stative verbs)

f) AGENT the performer of an action that affects another being.

g) COMPLEMENT inanimate or intangible things that come into being as the result of an action or process. They do not receive the effect of an action.

EX/ Build a bookcase. ACTION COMPLEMENT
Dust a bookcase. ACTION PATIENT

h) BENEFICIARY/RECIPIENT person or animate thing that profits or benefits from an action or process. Verbs such as "receive", "accept", "get" have beneficiaries.

i) POSSESSOR someone or something that owns or possesses something.

EX/ John has a parrot. POSSESSOR PROCESS PATIENT

j) PART parts of entities.

EX/ Dogs have paws. ENTITY STATIVE PART

k) ATTRIBUTE the subcases of modifiers. They include: size, age, condition, shape, color, quality(inherent characteristic), condition(not inherent and/or subject to change) see p 50 (VH).

l) EQUIVALENT a noun with a syntactic role as a predicate noun is equivalent to the entity.

EX/ Mary is a Nurse. Tom will become a doctor.
The apple turned to gold.
The girls stayed friends.
He remained a teacher.
John became a firefighter.

m) REASON/CAUSE see p 138-140 (VH)

n) TIME This thematic role appears to include the subroles of FREQUENCY and DURATION.

o) LOCATION The place of the occurence.

p) MANNER this thematic role includes the subrole of INTENSIFIER.

q) CONCESSION this role is pointed to by "although" or "though".

r) EFFECT/CONSEQUENCE

B. Sentence Patterns

1) Simple sentences - simple, active, declarative, positive (VH)

Formats:
- NP = Nounphrase
- Vi = Intransitive verb(doesn't need a NP or adj to complete its meaning)
- V = Mostly transitive verbs which allow a passive voice, plus "have", "cost", weigh".
- Vl = Linking verb that links an adjective to a NP. Some linking verbs (STATIVE) are: be, taste, feel, smell, look, sound, turn, grow, seem, become, remain, stay, appear
- Vbe = A form of the verb "to be".
- ( ) = Optional part
- Adj = Adjective
- Adv = Adverbial

a) Pattern 1 : NP + Vi + (Adv)
The adverbials allowed in Pattern 1 are those of location, manner, time, frequency, and duration.
The Pattern 1 particles are: down, up, around, out, in.

b) Pattern 2 : NP1 + V + NP2 + (Adv)
All Pattern 1 adverbials are allowed plus indirect object phrases. The most common transitive verbs that allow both direct and indirect objects are:

ask	buy	give	offer	read	send	teach	throw
bring	find	make	pay	sell	throw	tell	toss

Many Pattern 2 verbs have particles as complements. These are called "two word verbs" or "double verbs" in the literature. The particles are:
off, out, up, on, over, away, down, in

c) Pattern 3 : NP + Vl + Adj + (Adv)
Np is the subject of this sentence. In Pattern 3, you cannot use an adverbial of MANNER with be, remain, or seem. Pattern 3 adverbials are of time, location, duration, and frequency.

d) Pattern 4 : NP1 + Vl + NP1 + (Adv)
NP1 stands for the subject and its EQUIVALENT.
The linking verbs can include particle complements.
EX/ turn into, turn to, change to, change into.

e) Pattern 5 : NP + Vbe + Adv + (Adv)
The first adverbial is obligatory and is most usually location or time.

2) Compound sentences (RH)

Definition: A clause is a group of words containing a nounphrase and a verbphrase that is part of sentence.
Definition: A Main clause is a clause that can stand alone as a sentence.

A Compound sentence contains 2 or more main clauses. They are separated by conjunctions such as "and", "or", "but", "nor", or "so". In some cases, the conjuction is a comma or semicolon.

3) Complex sentences (RH)

Definition: A subordinate clause functions as a dependent clause within a larger construction that is itself a clause or a constituent of one. It cannot stand alone.

A Complex Sentence contains a main clause and one or more subordinate clauses.

2. Questions

A. List of categories

1) Open Questions (RH) or Wh-questions (VH).
The set of possible values for the variable is open or infinite.
ex: Who wrote the letter? -> What person(X) wrote the letter?
Person(X) can be : She or John or anyone.

a) Without Inversion (VH) - Substitution of WH word for variable found as the subject of a sentence or as a modifier of the subject. The Wh- words that can be used in this category are:
Who, what, whose, which.

Ex: Who is an astronaut? -> SUBJECT, AGENT
This question requires a nounphrase answer.

What is in the cargo bay? -> SUBJECT
Whose parts are used to build the shuttle? -> Modifier
This question requires a THEME or Subject modifier response.
Which shuttle flew the longest? -> SUBJECT

b) With Inversion (VH) - When information is contained in an optional role, Theme(Object), or pertains to an adverbial, copula (be verb) or auxiliary verb inversion must occur.
The WH- words that can be used in this category are:
What, when, where, why, how, and who (if theme).
EX: What is the crew doing? -> THEME
When is the crew leaving? -> TIME
Where is the launchpad? -> LOCATION
Why is the shuttle yawing? -> CAUSE/REASON
How can Bill study all night? -> MANNER, ACTION
Who make up the crew? -> BEMOD, THEME
- This question requires a nounphrase response.

c) With DO support (VH)
1) For the auxiliary inversion in simple present or past tense. The WH-words that can be used in this category are:
What, why, how, where, when, whose, which.

EX1 by evolution:

Statement	He broke (something).
DO support	He did break (something).
AuxInversion	Did he break (something/what)?
WH-question	What did he break?

EX2: John broke Mary's stick for the fire by bending it across his knee.

1.	What did John break?	THEME
	What kind of stick did he break?	modifier of THEME or Subject
2.	Why did John break the stick?	REASON/CAUSE
3.	How did John break the stick?	MANNER
	How many sticks did John break?	modifier of
	How much milk did you get?	THEME or AGENT
4.	Where did John break the stick?	LOCATION
5.	When did John break the stick?	TIME
6.	Whose stick did John break?	THEME Modifier.

2) Questions asking about the activity (action) are of the 'what . . . do' form and require a verbphrase or sentence response.

Ex: What did the astronauts do? <- DO support
What will they do now? <-|
What had the support crew done? |- have only aux
What is the astronaut doing? <-| inversion

d) With prepositions (VH) -

EX: info wanted	Question	Statement/answer
to/for+IO	To whom did he send the gift?	He sent the gift to (someone).

This question requires a nounphrase response stating the beneficiary.

location maybe destination?	To what city are they moving?	They are moving to (some city).
time	At what time is the meeting?	The meeting is at (some time).
reason/cause	For what did he buy the item?	He bought the item for (some reason).

These could also have been expressed using Who, Where, When, and Why.

e) Questions requiring a summarization or recounting of events begin with: 'What happened'.

Ex: What happened to the Challenger?
What happened after the Challenger blew up?
What happened at the control center?

f) Echo Questions (RH & VH) NOT TO BE CONSIDERED IN QUERY SYSTEM

Questions where the Wh-word is used for what the individual wants repeated for affirmation or wants to express surprise about.

EX: He did WHAT? The shuttle is leaving WHEN?
The shuttle is HOW tall?

g) Wh-questions requiring Adverbial Responses (VH)
The Wh-words that can be used in this category are: Where, when, why, how long(CONFUSING), how often, how, and how(adj).

```
Ex:  where              -> LOCATION
     when(what time)    -> TIME
     why(what...for)    -> REASON/CAUSE
     how long           -> DURATION
     how often          -> FREQUENCY
     how                -> MANNER
     how (adj)          -> BEMOD, or topic modifier
                             how big (tall,heavy,cold)
```

questions can begin with prepositional phrases except when containing an indirect object: In what city, at what time, to what city. . .

2) Closed Questions (RH)

a) Yes-No (VH) and (RH)
The set of answer values is yes or no. Ex: Is he finished?

b) Or (RH)
The set of alternative responses are in the question.
Ex: Is the shuttle in CA or FL?

B. Categories for each Wh-word

1) Who 1a, 1b, 1d

2) What 1a, 1b, 1c, 1e

3) When 1b, 1c, 1d, 1g

4) Why 1b, 1c, 1d, 1g

5) Where 1b, 1c, 1d, 1g

6) How 1b, 1c, 1g

7) Whose 1a

8) Which 1a, 1c

9) Whom 1d

C. Categorization of 35 questions (List attached)

	Question	Category
1)	What is the maximum cargo weight the shuttle can carry?	1a
2)	How far can the shuttle transport cargo from the earth's surface?	1g
3)	What has happened to the Enterprise?	1e
4)	How many years of education are required for astronaut candidates?	1c1
5)	What is the total weight of the shuttle?	1a
6)	How thick is the window of the shuttle?	1g
7)	How many gallons of liquid hydrogen fuel can the storage tank hold?	1c1
8)	What type of liquid fuel is used on the shuttle?	1c1
9)	What is the descent rate of the shuttle during landing?	1a
10)	How long is the machanical arm used for payload deployment?	1g
11)	What are the dimensions of the cargo area in the shuttle?	1a
12)	How is waste disposed of?	1b
13)	Have there been astronauts picked from minority groups?	2a
14)	What is the total number of times that the shuttle has been launched?	1a,b
15)	What type of food do astronauts eat during a shuttle mission?	1c1
16)	What is the orbiter's velocity while in orbit?	1a,b
17)	What is the maximum acceleration of the shuttle during launch?	1a,b
18)	What is the maximum touchdown glide speed of the shuttle?	1a,b
19)	How many pounds of thrust do the SRB booster rockets generate during liftoff?	1c1
20)	What is the maximum fluid fuel flow rate during launch?	1a,b
21)	How fast does the crawler or transporter travel?	1g
22)	At what altitude and speed must the pilot throttle back during ascent?	1b
23)	a) How many general purpose computers are on board the shuttle?	1c1
	b) What functions do they serve?	1c2
24)	What is the new design of general purpose computers like on board the shuttle?	1b

25)	What is the total number of tiles that cover the orbiter for thermal protection during reentry?	1a,b
26)	When did the first space shuttle launch occur?	1g
27)	How long is the runway at a shuttle landing facility?	1g
28)	What type of glass is used for the windows to withstand the pressure of flight	1c1
29)	What is the total amount of RAM available in the shuttle's general purpose flight computer?	1a,b
30)	What material are the heat shield tiles composed of?	1b
31)	What type of computer guidance and navigation systems does the shuttle use during reentry and landing?	1c1
32)	What is the maximum power available to the payload area?	1a,b
33)	Are there emergency escape prodedures to jettison the crew members out of the shuttle?	2a
34)	Where do the crew members sleep on the shuttle?	1g
35)	What is the color of the external tank?	1b,c

D. Question Type Count

1a	1b	1a or b	1b or c	1c1	1c2	1e	1g	2a
4	4	8	1	8	1	1	7	2

E. Question Types in TRQS (JD)

	Category
1) Query what can be asked Cue words: Subjects or Topics. Answer: All Subjects and Objects in target database.	1b
2) Query the existence of a fact Cue words are: Who, Which, Whom, and What. Answer: Complete sentence.	1a,b,c,d,e
3) Query description in detail of an event. Cue words are: Who, Which, Whom, and What. Answer: Complete sentence.	1a,b,c,d,e
4) Query LOCATION of an event Cue word: Where. Answer: Location Thematic Role.	1b,c,d,g
5) Query SOURCE of an event Cue word: Where. Answer: Source Thematic Role.	1b,c,d,g
6) Query the DESTINATION of an event No cue word or answer set that will return the Destination Thematic Role.	
7) Query the TIME of an event Cue word: When. Answer: Time Thematic Role.	1b,c,d,g
8) Query the person or thing that causes an event Cue words are: Who, Which, Whom, and What. Answer: Complete Sentence.	1a,b,c,d,e
9) Query the INSTRUMENT or CONVEYANCE of an event Cue words are: Who, Which, Whom, and What. Answer: Complete Sentence.	1a,b,c,d,e
10) Query the BENEFICIARY of an event Cue words: Benefited From. Answer: Beneficiary Thematic Role.	1a,b,d
11) Query the PURPOSE of an event Cue word: Why. Answer: Purpose Thematic Role.	1b,c,d,g

12) Query the DURATION of an event 1g
Cue words: How Long.
Answer: Duration Thematic Role.

13) Query the CO-AGENT of an event 1a,b,c,d,e
Cue words are: Who, Which, Whom, and What.
Answer: Complete Sentence.

14) Query the event which happens to the AGENT or OBJECT 1a,b,c,d,e
Cue words are: Who, Which, Whom, and What.
Answer: Complete Sentence.

Bibliography:

(JD) .. Driscoll, James R. A Portable Natural Language Query Interface for Databases Containing Large Amounts of Textual Data. Manuscript submitted to 16th Very Large Data Base Conference.

(VH) .. Heidinger, Virginia T. Analyzing Syntax & Semantics. Washington, D.C.: Gallaudet College Press, 1984.

(RH) .. Huddleston, Rodney. English grammar: an outline. Cambridge, GB: Cambridge University Press, 1988.

Section 6.

Manuscript about the SPIRIT text retrieval system

from August 1990 DATABASE Magazine

Page intentionally left blank

NEW DIRECTIONS FOR MICROCOMPUTER-BASED HYPERTEXT SYSTEMS

by Kenneth Ray
and
James R. Driscoll

In the second half of the twentieth century major advances have been made in several fields relevant to information management. Among these are the continued development of the microcomputer, high-level programming languages, and sophisticated methods of textual analysis. It is estimated that as of 1985 there were 1.7 billion documents online worldwide [1], and although powerful personal workstations are increasingly common on the desks of professionals, it is a sad reality that even with access to large, continually updated databases of relevant data, most of these professionals do not have the software tools required to effectively locate and organize the available information.

At present, information retrieval (IR) activities are generally conducted using the interfaces provided by commercial databases, under the control of professional search intermediaries or end-users who possess extensive training. These database systems typically reside on mainframe computers and lack the user-friendly interfaces which personal computer users have come to expect. It seems a reasonable goal to build effective, easy to use IR systems which allow workers to focus directly on their problems and tasks, without resorting to intervening technicians or programming languages. This article describes an advanced hypertext environment which couples a probabilistic and linguistic approach to information retrieval with the intuitive, easily browsable document representation characteristic of conventional hypertext systems.

(Editor's Note: Readers needing a refresher on conventional hypertext systems should read Carl Franklin's article, "Hypertext Defined and Applied," in the May 1989 issue of ONLINE *13, No. 3, pp. 37-49. —PH)*

COMMERCIAL INFORMATION RETRIEVAL SYSTEMS

One of the reasons people do not find what they are looking for in books or textual databases is because the terms they use to describe the things they want are not the terms indexed by the system. In a commercial IR system, based on a Boolean model, this failing can usually be attributed to imprecise or incomplete search terms. It is well known that Boolean queries are not only difficult to construct, but require a trade off between recall (the proportion of appropriate references found) against precision (the proportion of references which are relevant). In practice, a compromise is often obtained by formulating a query which is neither too broad nor too narrow, resulting in a large set of candidate documents, only a few of which will be relevant to the searcher's needs. In general, it is not a good idea to impose Boolean search techniques on a microcomputer user who has no experience in the trade offs required and who may be unwilling or unable to formulate complex queries.

Fortunately, commercial IR systems are now available which have the ability to process natural language queries through probabilistic and linguistic analysis. The information in the textual database of such a system is represented as a collection of index terms which can be matched against a processed query. Those documents whose representations most closely match the query are returned in the form of a ranked list of document classes. This approach has several major benefits. For one, users do not require extensive training as to the system's use, thus freeing them to focus on the task which motivated them to use an IR system in the first place.

These systems employ a hypertext document representation which allows users to browse through the documents gathered in response to a query. The user may choose to read the documents

in a serial fashion, as they were written by the author, or to leap from relevant passage to relevant passage between documents, or to formulate a query from any page of the current document being browsed. When implemented through the intuitive interface typical of conventional hypertext systems, this last ability becomes the basis for a relevance feedback loop. Through successive iterations of query reformulation, the user can "home in" on the collection of documents which best suit his or her interest.

LIMITATIONS OF CONVENTIONAL HYPERTEXT SYSTEMS

The nodes of information which make up a conventional hypertext document are traditionally linked by hand. This is a time consuming task which requires much thought. For applications which use large, dynamic collections of online documentation, it is impractical to attempt the construction and continuous adjustment of a conventional hypertext representation. In contrast, the set of ranked documents gathered in response to a user query on hypertext-based IR systems represents a hypertext structure which requires no maintenance.

Even when a conventional hypertext document has been well designed, it is possible to get lost while browsing through unfamiliar information. While a good author in a paper document will guide the reader through a progression of relevant points, hypertext gives readers the power to guide themselves. The double-edged sword of freedom of movement is both the beauty and the pitfall of conventional hypertext systems. Hypertext-based IR systems solve this problem through the automatic generation of user-defined links, represented by the keywords common to a set of documents. Because the design of this hypertext structure is instantaneously responsive to the will of the framer of a query, the question of where you are in someone elses hypertext network never arises.

ADVANCED HYPERTEXT SYSTEMS

We define an advanced hypertext system as an information retrieval system which processes natural language queries, automatically generates cross-referential hypertext links, includes a relevance feedback system, and maintains the easily browsable document representation characteristic of conventional hypertext systems. We will now discuss a commercial IR product called SPIRIT which represents the basis of such a system.

The SPIRIT system (Syntactic and Probabilistic Indexation and Retrieval of Text) is available through SYSTEX, a French company founded in 1979 by a group of researchers in order to make industrial products in the fields of computational linguistics and artificial intelligence, applied to information retrieval. The SPIRIT system is multilingual with versions in English, French and Arabic. A microcomputer-based, English version of SPIRIT which runs under OS/2 or UNIX should be available by the time you receive this issue of *DATABASE*.

SPIRIT processes natural language queries by computing the semantic proximity between a query and the contents of an indexed textual database, using both statistical and linguistic analysis. SPIRIT does not try to understand a query or the database text; instead, SPIRIT examines each word in a query and database text considering word frequency, word position, and linguistic word forms (e.g., part of speech). The system establishes an ordered list of candidate documents based on their semantic proximity to the query. The linguistic processing of queries includes spelling correction, idiomatic expression recognition, synonym recognition (e.g., "lorry" and "truck" are considered identical), and the resolution of grammatical ambiguities (e.g., "can" as auxiliary verb is distinguishable from its use as a noun). One well-formed natural language query over a selected database can obtain the same result as a complete Boolean search strategy of many Boolean queries. Because natural language is used to express queries, professional search intermediaries are not needed to translate user queries into a form compatible with the indexed terms of a database.

One well-formed natural language query over a selected database can obtain the same result as a complete Boolean search strategy of many Boolean queries.

SOFTWARE PROFILE

SPIRIT

SYSTEX Company
FERME DU MOULON
91190 GIF SUR YVETTE
FRANCE
Telephone: 33 (1) 69 85 33 38
Fax: 33 (1) 60 19 13 12
From the USA it is best to contact SYSTEX via fax.

Cost: from 55,000 FF (single user) to 150,000 FF (multiuser) to 500,000 FF (mainframe)

Hardware/Software Requirements (any of the following systems):

- IBM mainframe using VM or MVS operating system.
- VAX computer using VMS.
- SUN or other workstations using UNIX.
- IBM AT class microcomputers using OS/2 or UNIX.
- Macintosh (available in late 1990).

The version of SPIRIT (2.1) with which the authors are familiar is installed on an IBM 4381 mainframe at the University of Central Florida, and although it does not incorporate synonym recognition, we have been quite impressed with the system's performance in rapidly retrieving relevant text. However, because this version is installed on a mainframe, accessed through character-based terminals, the system screens are not as pleasant as those normally seen on a personal computer.

The indexing of documents in a database is the most time consuming task of using the SPIRIT system. All documents must first be placed in one file (ASCII for microcomputers, EBCDIC for IBM mainframes), with markers separating each document. For example, in demonstrating the SPIRIT system to NASA, Kennedy Space Center, the one thousand page *NASA National Space Transportation System Reference* (better known as the shuttle manual) was marked by considering each paragraph of the manual as a document. This resulted in a database of 4902 documents which required 3.5 hours to index [2]. Of course, a database is only indexed after new material is entered.

FIGURE 1
INITIAL MENU

```
SPIRIT                6.7 R 0.0              - VM / CMS        MENU 0

                1.  SAVING/STORING DOCUMENTS
                2.  CREATE A DATABASE
                3.  UPDATE A DATABASE
                4.  DELETE SOME DOCUMENTS
                5.  DETECTION OF ERRORS

                S.  QUERY THE DATABASE

                G.  MANAGEMENT UTILITIES
                P.  REDEFINE DISK PARAMETERS
                V.  EXECUTE A VM COMMAND
                F.  END

       BY DEFAULT (ENTER) => QUERY THE DATABASE
```

FIGURE 2
MAIN MENU

```
                 ********************************
                 *                              *
                 *           SPIRIT             *
                 *           SYSTEM             *
                 *            R 2.1             *
                 *                              *
                 ********************************

                      PRINCIPAL  MENU
                     ________________

MENU: (QUERY,AFQUERY,CONTQ,BOOL,DOCQ,ANSWER,DOC,BASE,STOP,PRINT,G,?):
```

QUERYING THE SHUTTLE DATABASE USING SPIRIT

SPIRIT is menu-driven and, once set up, is very easy to use. The initial system screen is reproduced in Figure 1. The first five options on the menu provide various database development and maintenance functions. The **S** option of the initial menu allows you to query SPIRIT on the information contained in a database. With the exception of **F**, which terminates the program, the remaining options provide environment specific utilities.

After choosing **S** from the initial system menu, the next SPIRIT screen lists the various databases within the system, the date they were last updated, and prompts you to enter the name of the database you wish to query. Entering a database name causes the main menu to be presented (Figure 2).

The system commands are as follows:

- **QUERY** is used to form a natural language query on the textual fields of a database.
- **AFQUERY** is used to form a natural language query on all fields of the database, not just the textual fields.
- **CONTQ** is used to continue a query, make more than one query simultaneously, or to alter a query.
- **BOOL** is used for Boolean queries.
- **DOCQ** allows all the keywords within a chosen document to be used to query the database.
- **ANSWER** shows the list of documents which satisfied the last query.
- **DOC** is used to choose a document to browse.
- **BASE** is used to change databases.
- **STOP** ends the session and returns to the principal menu.
- **PRINT** is used to print documents.
- **G** (rid queries) allows queries on multiple fields.
- **HISTO** displays a list of the previous questions in a session.

We choose the **QUERY** option from the main menu and enter the question **What is the maximum cargo weight the shuttle can carry?**. After we do this, the screen appears as shown in Figure 3. All questions to the system must end with a question mark, and if SPIRIT detects a word within a query which is misspelled, you will be prompted to reenter the word. The empty words shown in Figure 3 will be discarded while the keywords will be used to formulate a query.

Whenever *** appears on screen it signals the user to press the return key to continue. A few seconds after pressing the <return> key, a ranked list of document classes appears, as shown in Figure 4. SPIRIT ranks the classes by the number of keywords matched, the proximity of the keywords to each other, and other statistical and linguistic factors. Keywords in close proximity are separated with a hyphen (-). Notice that SPIRIT considers the two keywords in CLASS 2 more relevant to our query than the four

FIGURE 3
NATURAL LANGUAGE QUERY

NATURAL LANGUAGE QUERY ON THE SHUTTLE BASE

<1>: What is the maximum cargo weight the shuttle can carry?

EMPTY WORDS : what, is, the, the, can.
KEY WORDS : maximum, cargo, weight, shuttle, carry.

***

FIGURE 4
DOCUMENT CLASSES

CLASSES	NB DOCS	KEY-WORDS
1	1	maximum-cargo-weight.
2	1	shuttle-carry.
3	2	cargo-weight, shuttle.
4	2	cargo, weight, shuttle, carry.
5	3	cargo-weight.
6	1	cargo, weight, carry.
7	2	cargo, weight, shuttle.
8	6	cargo, shuttle, carry.
9	1	weight, shuttle, carry.
10	1	maximum, weight.
11	4	cargo, weight.
12	4	cargo, carry.
13	6	maximum, weight.
14	2	maximum, carry.
15	20	cargo, shuttle.
16	9	weight, shuttle.
17	13	shuttle, carry.
BOTTOM OF LIST		

LIST OF CLASSES TO BE DISPLAYED (?) : 1

FIGURE 5
DOCUMENT DISPLAY

DOC 3976 BASE : doc 3976 NCP:0/CPI:1/NBI:1 +18 1K/1K
IDENTIFIER. : doc 3976
TEXT....... :

Cargo weight is defined as the payload control **weight** plus the **weight** of the attached hardware used to secure the payload to the orbiter. Allowable **cargo weight** is determined by altitude and orbital inclination. For example, on a standard inclination of 28.45 degrees, **maximum cargo weight** capability in a circular orbit at an altitude of 100 nautical miles is about 55,000 lb. This capacity decreases with altitude and falls to about 40,000 lb. in a 300-mile circular orbit. At the higher inclination of 57 degrees (also a standard inclination), **cargo weight** capability is 40,000 lb. in a 100-mile circular orbit. This decreases to slightly over 20,000 lb. in a 320-mile orbit. These **weights** are those for a nominal ascent for what is described as a "simple, short duration, satellite deploy mission."
BOTTOM OF DOCUMENT

INFORMATIONAL PAGE 1/1
WHAT DO YOU WANT TO DISPLAY ?
\> OR RETURN, <, >>, <<, DOC, END, DDQ, (?) :

FIGURE 6
DOCUMENT DISPLAY

DOC 3976 BASE : doc 3976 NCP:0/CPI:1/NBI:1 +18 1K/1K
IDENTIFIER. : doc 3974
TEXT............ :

For Vandenburg Air Force Base western test range **satellite deploy missions**, using OV-103 or OV-104, the **cargo-lift weight capability** is 29,600 pounds for a **98-degree** launch **inclination** and a **110-nautical-mile (126-statute-mile)** polar **orbit**. Again, an increase in **altitude** costs approximately 100 pounds per **nautical mile.** NASA also assumes that the advanced solid rocket motor will replace the filament-wound solid rocket motor case previously used for test range assessments.
BOTTOM OF DOCUMENT

YOU ARE IN DDQ
WHAT DO YOU WANT TO DISPLAY ?
\> OR RETURN, <, >>, <<, DOC, END, DDQ, (?) :

keywords in CLASS 4, and that the combination **shuttle-carry** is more relevant than **cargo-weight**.

One selects the document classes to be browsed by entering a number or list of numbers separated by commas. In our case, the document which contains the keywords **maximum-cargo-weight** is considered to be the most relevant, so we simply enter a **1**. SPIRIT responds as shown in Figure 5.

All keywords are highlighted within the page currently being browsed. The prompt shown below the document includes options for further browsing through the classes of documents in Figure 4, or within the pages of the current document (page up, page down, first page, last page). In our case, we quickly note that this page answers our question and further browsing is not necessary. By pressing **E** for end, we will be returned to the main menu.

BROWSING A DATABASE

On the other hand, if we decide to browse through the entire collection of documents, the repeated pressing of the <return> key will page forward through the current document class, retrieve the next ranked class, and so on until all classes requested have been viewed. If at any time during the browsing of a document class, we encounter a page of information which interests us, but is not directly related to the material our query has gathered, we can employ the **DDQ** option. This option allows the reformulation of a query based on all keywords contained in the current page being viewed.

For example, after reading the paragraph retrieved in Figure 5, suppose we become interested in the effect of altitude and inclination on cargo weight capacity and desire more information relevant to this paragraph. By selecting the **DDQ** option, SPIRIT automatically links us to the most relevant paragraph in the shuttle manual. The result is the screen displayed in Figure 6. Notice that this paragraph provides additional information on the topics which interest us. This represents the most relevant document in a new collection of ranked documents, which can be browsed in the same manner as the previous collection.

CONCLUSION

In a society approaching the state of information overload, it is becoming increasingly apparent that our present methods of retrieving and organizing textual material are too primitive to keep up with the world-wide pace of technological development. The authors believe that the development of hypertext IR systems is an important step in the evolution of information management tools, and that the SPIRIT system embodies the basic requirements of what we define as an advanced hypertext system.

REFERENCES

[1] Williams, M. W. *Proceedings of the Eighth National Online Meeting*, New York, NY. (1987).

[2] Clark, M. A. "A Prototype Text Retrieval System for Answering NASA Kennedy Space Center Media Questions." Research project report submitted to the Department of Computer Science, University of Central Florida, Orlando, FL. (December 1989).

THE AUTHORS

JAMES R. DRISCOLL is an Associate Professor of Computer Science at the University of Central Florida. He has a B.S. degree in electrical engineering and a M.S. and Ph.D. degrees in computer science, all from the University of Kansas. He has published eighteen manuscripts in the last four years in regard to efficient file organization and automatic text processing (automatic keywording, text retrieval, and intelligent text manipulation).

For the past three years he has been continuously funded in these areas of research by the Department of Defense, the Army Research Institute, and NASA. He is a member of the Association for Computing Machinery and the IEEE Computer Society.

KENNETH RAY is a software engineer for Martin Marietta Information Systems in Orlando, Florida.

Communications to the authors should be addressed to James R. Driscoll, Department of Computer Science, University of Central Florida, Orlando, FL 32816; 407/275-2341.

NASA/KSC Cooperative Agreement

MEMORY-BASED REASONING FOR ADVANCED LAUNCH SYSTEM OPERATIONS

1990 Annual Report

Dr. Harley R. Myler, Principal Investigator
Department of Computer Engineering
University of Central Florida
(407) 823-5098
hrm@engr.ucf.edu

Dean A. Dubois, Research Assistant
Department of Computer Engineering
University of Central Florida
dad@engr.ucf.edu

January 25, 1991

I. Introduction

This project addresses a need for investigation into advanced techniques of automated launch processing. Our thesis has been that Artificial Neural Networks (ANNs) are a prime candidate for the architecture of a launch processing system that would use a Memory-Based Reasoning paradigm (Stanfill & Waltz, 1986) in order to observe, evaluate and suggest corrective action during a launch sequence. It should be clear that the sequence of events that lead to the successful launch of a space vehicle or other payload is a complex, time-dependent process. In addition, it is a process that is highly dependent on human capability and attention. It is important to note that we seek to produce an automated launch paradigm that would relieve a portion of the human attentional burden from the launch personnel.

The long-term objectives of this project have been to:

- Create a model scenario of Launch Operations subsystems.
- Generate an I/O structure for the model.
- Generate an information exchange protocol for the I/O.
- Build an appropriate neural model for reasoning on the activities of the model scenario.
- Test the model.
- Extend the model to large parallel processing hardwares.

A necessary objective of this research is to develop a general paradigm of automated reasoning based on memory associations. At present, we have accomplished the first four objectives and are working on a model system to run on a parallel computer purchased for this project. The details of our model and the parallel processing system are discussed in the body of this report.

II. Memory-Based Reasoning for Advanced Launch Operations

Memory-Based Reasoning (MBR) is reasoning that takes place as a function of data retrieval from a memory system. It has been discussed at length in previous reports published for this project and at presentations. Additionally, a paper entitled "A Neural Model for Memory-Based Reasoning in Advanced Launch Operations" is currently at press (Myler, 1991) that discusses and illustrates some basic and early results of this research. The concept of using MBR as a mechanism for automated launch monitoring was initially suggested in the proposal submitted to fund this research. In addition, the use of an artificial neural network (ANN) as the associative store for the system is also a unique invention of this research. Two problems arise when working with both MBR paradigms and ANNs--the problems of data formatting and data interpretation. In addition to these problems, we have

uncovered an additional impediment to the development of a system in the problem of representing and evaluating temporal sequencing.

In the case of data formatting, the immediate question is, "What form should the data take as an input to the neural system such that a correct interpretation of the output resulting from that data can be made?". Initially, our proposition was that the MBR system would work as a high level executive and intelligently process and analyze data from sophisticated sources. These sources would include automated subsystems imbued with their own reasoning capabilities, yet restricted to specific domains. For example, weather processing, scheduling, logistics operations, and process systems diagnosis and control. Automated reasoning systems have been researched by NASA to address all four of these areas. An early version of this project, the Advanced Launch Operations Intelligent System (ALOIS), was modelled using a hypermedia development tool and illustrated how intelligent subsystems might interact and communicate with each other.

Originally, the problem of data formatting was to be solved by use of hypermedia as a data constraint mechanism. The four systems to be intelligently integrated would interact with the MBR system via high-level indicators and responses to commands. In our demonstration, this interaction was accomplished via on-screen, mouse selectable buttons and text fields. Liberal use of animated graphics and digitized and synthesized sound added to the effect of a well choreographed interaction of subsystems. Our eventual intent was to replace the fixed decision scripts (the hypermedia program) with a neural system that would react using memory associations of event order. If event order or occurrence was inconsistent with the existing memory contents, the system would react with an anomaly indication. The overriding approach was to be a system that would integrate a large database of knowledge (of launch system processes) and compare it to an existing state of the monitored system. The system then react by drawing attention to an anomaly response developed from a comparison of the current state to a state that it had previously observed. In short, the system would learn from analogy and example. The setup conditions for learning to take place is determined from how the data interface of the hypermedia system has been formulated.

The hypermedia approach to MBR is similar to that proposed for the *Knoesphere* system (Lenat, *et al*, 1983), a proposed hypermedia-based expert system with encyclopedic knowledge. The *Knoesphere* is intended to be frame-based, with heuristic rules that govern the building and checking of user models (the system is intended to adapt itself to a user's ability level) and the decision of which modalities to activate based on the data presented. The difference between the MBR system and the *Knoesphere* is that *a priori* knowledge is only supplied as a function of the hypermedia interface structure. The actual knowledge data is completely unstructured and exists only as weights within the neural sponge of the

ANN used for the associative store. Our conjecture is that the system must learn it's knowledge in the way that a human does and that this knowledge cannot be injected explicitly. A great deal of discussion has addressed this statement (Dreyfus, 1979; Dreyfus & Dreyfus, 1988; Minsky, 1985) and we will not belabor it here. Of greater importance is the validity of our conclusion that a neural system is capable of both learning and associating knowledge intelligently.

We draw some measure of confidence that machine learning can take place from studies in neurobiology. Learning and conditioning are thought to be encoded in the neural subsystems of all animals possessing nerve cells (Kanigel, 1987). We must ask the question, "How does a single neuron, an elementary computing element of extreme simplicity, contribute to the incredible knowledge processing power that a brain possesses?". The answer to this is not in the individual neuron, but rather in the collective behavior of huge numbers of neurons and their interconnectivity. Of importance is our understanding of the complexity and immensity of the human brain, our only model of advanced intelligence. To underscore this, we quote:

> One peculiarity of biological neural networks is their size: in the whole human central nervous system there are on the order of 10^{11} neurons, but the number of their interconnections is still higher, probably up to the order of 10^{15}. It does not seem posssible to *program* the function of such a system according to a prior plan, not even by genetic information; consider, too, that the size and structure of the network is changing radically during and after childhood, when it is already in use (Kohonen, 1988).

If one is to program a neural system, then this programming must take the form of a change to the interconnectivity structure of the network either in terms of the mapping of interconnects or the signal weights between the neurons. Nature has had the advantage of billions of years of evolution to accomplish the complex structural programming of animal neural systems. In order to approach a limited emulation of this programming, we must content ourselves to focussing on the external data formatting of the information to be stored and associated.

To summarize, an MBR system associatively stores information and exhibits a reasoning behavior by making inferences from previous experiences of data. The simplest example of this is the following scenario:

assertion: The scheduled **launch** is in **three hours**.
assertion: The weather forecast indicates **bad weather** in **three hours**.
conclusion: **Launch cannot take place** with **bad weather**.

It is easy to see how a rule that operates on comparing the predicted weather conditions with the time of anticipated launch could easily reach the correct

called) is capable of modelling virtually any type of neural network to virtually any size depending on the capabilities of the computer host. The first topology proposed for MBR was a three layer fully connected network, similar to that shown in Figure 1. The input layer data consisted of four fields of five characters going to individual nodes. The middle, or hidden, layer had five nodes and the output layer had an identical structure to the input layer. Groups of characters were chosen to represent various conditions that a space shuttle awaiting launch might encounter. These data elements were determined in an *ad hoc* fashion from our, albeit limited, knowledge of launch operations. It must be emphasized that our goal was to examine the functionality of the model in a launch processing context and not to simulate an actual launch scenario.

Abbreviations were selected to indicate the status of WEATHER, SCHEDULING, and SYSTEMS and insure that no ambiguities existed between data elements. Each of these inputs is meant to simulate the input from automated reasoning sub-systems that evaluate conditions within their realm of understanding and report to the launch director. The possible conditions that can be reported by each of the input fields is:

Weather:
1) CLEAR - clear skies (good visibility is required for a launch)
2) THSTO - thunderstorm (may require mission to be scrubbed)
3) FREZE - freezing conditions (restrictions to operational temp)
4) RASTO - rainstorm (fowl weather forces a delay in the count)
5) CLODY - cloudy (visibility in case an emergency landing must be made)
6) FOGGY - foggy (altitude of cover effects documentary filming)

Scheduling:
1) NODEL - no delay (all systems ready)
2) SHDEL - short delay (required for system readiness)
3) LNDEL - long delay (major time delay evident)

Systems:
1) GOODY - all systems functioning
2) BDRED - bad reading (intermittent or faulting input)
3) BDELE - bad element (diagnostic failure)

The fourth input field indicates the launch status:

1) NOHLD - no hold (count proceeds uninterrupted)
2) IDHLD - indefinite hold (mission countdown suspended)
3) SCRUB - scrub (mission canceled)

Once the topology and data structure had been established, various forms of back propagation learning algorithms (see Carpenter, *et al*, 1987) were used to train the network. Training of multiple input/output pairs allowed all of the patterns that had been developed to be presented for training automatically. During this automated training mode, the maximum number of iterations to search for a converged network as well as the maximum error tolerance were entered. These

two parameters (along with the complexity of the network itself) determined the amount of time it would take to train the network. The results of multiple training runs of the initial topology are tabulated in Appendix A. This data shows an exponential relationship between the number of patterns that a network is expected to learn and the number of passes that are required to train those patterns. While the time to train the network may not be a priority issue at this time, the turn-around time for testing a wide variety of network configurations requires that the time for a network to successfully converge be kept at a minimum.

In an attempt to reduce the complexity of the network, the topology was changed to fifteen inputs (three fields of five characters each), a middle layer of five nodes and an output layer of five nodes. This topology allows only conditions of WEATHER, SCHEDULING and SYSTEMS to be input to the neural network, with no fourth data field for launch status. The output field is concerned only with hold/no hold status decisions. Under this scheme, the training time for the network was drastically reduced and the number of patterns that could be successfully converged upon was more than doubled.

Once the initial test patterns had been trained, a search for signs of emergent cognitive activity was pursued. New input patterns were presented and the computed outputs were observed. The new input patterns consisted of slight spelling alterations of the original patterns as well as new combinations of the input fields. The slight errors in spelling often resulted in an output pattern that was exact or very close to the original training pair. New combinations, however, yielded partial string responses with some of the characters differing from the trained pairs by as much as three hundred percent. These unexpected differences indicate that the generalizations formed by the network involve strong inter-field relationships.

In order to verify this speculation, as well as develop new techniques for training complex networks, several small subnetworks were designed. These subnetworks were to be trained to provide logical internal representations and then be combined together to form a single network. In order to test subnetwork performance without creating a network topology specifically designed to solve this problem, the initial format of the inputs was retained. Changes in the structure of the overall network were made by introducing small subsystems into each data field of the input stream. In short, a neural network was created for each input field. Each subnetwork was then trained to read the ASCII characters that could be entered in its respective input field and associate that character group to a set of numeric parameters. In the case of SCHEDULING, there was only one parameter, time. For the SYSTEMS input field, both a diagnostic parameter and a faulty reading parameter were created. The WEATHER subsystem required the greatest diversity resulting in eight parameters. The eight datums for weather were derived

from local climatological observations. Four fundamental elements were included for each possible weather condition. Visibility, temperature, humidity and cloud cover were encoded as two values each. A parameter for the mean value and one for the deviation of each of the elements for the four weather conditions were computed. The subnetworks were created and trained individually and training results for each subnetwork as well as the climatological data are presented in Appendix B. The resulting networks may be thought of as a dictionary, a network that has not been specifically trained with any knowledge of the over-all problem to be solved. Instead, these networks are able to read input characters and relate them to unique parameters. This conversion from lexigraphical data to numerical approximations should allow the system to read and write the character strings it has been trained, as well as form associations of known parameters. These associations should be much easier to identify and verify within the network's operation than the unknown relationships between the arbitrary characters of the input and output fields.

The three subnets were combined to form a single network and initial weights were set according to the results of the individual trainings. Because this network consisted of three discrete portions (to take into account the three distinct input fields), the new network topology, Figure 2, does not appear to be fully connected between successive layers. The connections still exist, however, their weight has been forced to zero. As long as the learning factor for this layer is kept at zero, there will be no interaction between the characters of these input fields. The original input patterns were presented with the learning factor for the first layer (the subnets or dictionary layer) set to 0. This allows the training previously recorded to remain while inferences are drawn from the new parameter space. The accumulated training time is much less for systems that are made up of subnets. Since the subnets are generic in nature, they could be used in multiple network designs without being retrained. The possibility of libraries of subnets with understanding in a wide variety of areas would allow the creation of complex networks by combining high-level components into new and unique configurations. The subnet approach also gives rise to easy implementation in a parallel machine.

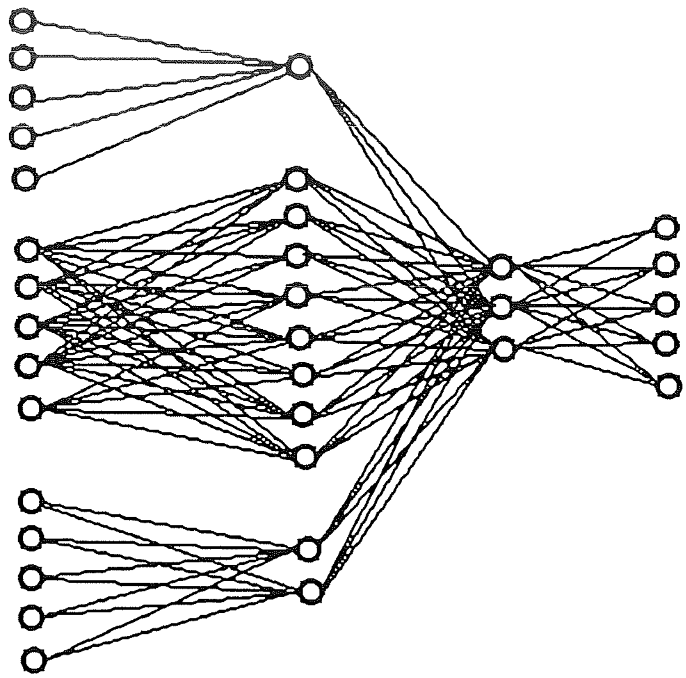


Figure 2. Subnet Network Topology

While this approach does not provide direct proof that an abstract correlation will be found by a neural network, it will allow us to apply known generalizations and measure quantitatively the success of the neural network in finding that relationship. Once this *level of learning* has been established, it should be possible to compute the likelihood that the system will find unforeseen correlations between input data. The success of this approach will also lead the way to modular construction of complex neural networks. Components may be separated and trained and then assembled into larger systems with all of the knowledge of the combined subsections without the high cost of training an entire blank system with equal capabilities.

Upon successful completion of this preliminary implementation of neural network launch system modeling, a more complex approach was developed. This new system took the results of earlier networks and tested the expandability of the concept. Expansion took two major forms, first, the number of nodes in the system was increased. This meant a much larger number of inputs to the system had to be tested. The second form of expansion required that the network respond within the context of a launch timing model and be temporally guided.

The expansion of the number of inputs to the system forced the creation of tools to specify, develop and edit large amounts of input data. It was determined that this data would be collated in the form of a DATA-BUFFER, and that an additional file (the DATA DICTIONARY) would be accessed in order to fully

describe the significance of the various portions of the data buffer. Finally, two programs had to be created to interpret the data buffer. The first program allowed the buffer to be modified both in raw hexadecimal form and in a high level manner. The second program converted the DATA-BUFFER into a file that could be presented to the neural network.

The complexity of incorporating the concept of time into the network presented the greatest challenge to this phase of our study. In order for the system to make intelligent decisions about the viability of a given launch scenario, data must be available to indicate the current state of the system, as well as the scheduling constraints. In essence, the launch model must maintain two timing criteria: a *real-time* schedule (which dictates launch window and weather prediction), as well as *T-time* (which represents the relative phase of the launch process). Each of the timing inputs to the network were translated into discrete time intervals and these representations are shown in Table 1. The network must be able to combine the information about launch window, weather conditions and launch status to form a complete picture of the possibility of launch. It should be noted that the time divisions for weather and launch phase are arbitrary boundaries while the launch window is specified in uniform increments. The ability of the network to combine these two forms of time representation should further prove the flexibility of the neural network system.

The *launch window* time is specified using 100 hours of yes/no inputs. This allows a flexible description of multiple windows of varying widths. The *weather prediction* is broken into four categories:

long term	40 hours or more	poor
mid term	6-39 hours	fair
short term	5 hours or less	good
current readings	NOW	excellent

The "T-" time is divided into six segments that span T-100 (shuttle preparation) to T+10 (shuttle clear of atmosphere):

T-100	T-40	T-2	T-10m	T-10s	T+10m
prep/load	roll out	board	fuel	ready	launch

Table 1. Network Timing Inputs

Once the timing had been incorporated into the network scheme, new forms of data had to be devised and described. The new data was presented from three categories: Weather, Systems and Command. The weather system represents the output of an automated reasoning system that is providing weather prediction and current conditions. The weather status is divided into four fields and each of these fields indicates a potential problem to a launch. The fields selected are: visibility,

wind velocity, lightning and temperature. Long, mid and short term predictions are provided as well as the current conditions for each field.

The Systems data consists of various valves and switches throughout the shuttle and support systems and indicate internal monitoring . Provision for bad readings has been made so that a failure may be simulated in any component or group of components. Additionally, redundant components may be bypassed. The decision to bypass or replace a component comes from the Command section. This final portion of the data represents the top of the launch decision ladder. Here, the next phase of the launch is determined and the system may either continue from one phase to the next (a normal launch), repeat the current phase (a hold), jump back to a previous phase (an extended hold) or scrub the launch. These inputs constitute the training data for the neural network. The conjecture here is that the system will learn to conduct an optimal sequence toward launch.

The project has reached a point where it has become necessary to expand the scope of the neural networks function. To this point, all input have produced an immediate response. Although this will allow the system to attack a large number of problems, it eliminates any concept of time-dependent functions. The next phase of the study will require that the system be able to relate stimuli and the state of the system to a time element. With the exception of one publication (Grossberg and Schmajuk,1989), this new category of problem has, to our knowledge, been virtually unexplored in the neural network field. This problem most closely resembles circuits with memory vs. circuits without memory in that a memoryless circuit need maintain (nor has the ability to do so) no knowledge of past events.

The following proposition must be addressed: Is the main distinction between a system that realizes time-dependent action and a direct stimulus/response system the existence of memory? On the broader scale, time is merely another dimension to any equation. Given the fact that a neural network is capable of learning a sufficiently complex equation, it follows that such a system should be able to create time-dependent functions. In this form, the concept of time would be introduced to the system as a single input and this value would effect all other inputs. If on the other hand, one were to introduce the concept of short-term-memory to a system, time dependent functions could also be resolved and the system would more closely resemble the human mental model. Under this scheme, time would be represented as the memory of the previous state of the machine and the combination of previous state and current state would create a stimulus for action.

Short-term memory is recognized as an important element in the human mental process(see Shiffren & Atkinson, 1969; Waldrop, 1985). Various forms of the short term memory paradigm are used in both the learning process and the

actual gathering of information for the brain. For instance, the response of the eye to light as well as the ear to sound are short-term integrating functions. The intensity of a given input is perceived as a function of both the actual pressure of the input and the duration of that pressure. These short term memory effects serve to synchronize the outside world with the relatively slow operating frequency of the brain. During the cognitive process, the brain keeps differing levels of knowledge. These different areas represent new versus old information that has been correlated into a set of rules. The different areas of knowledge have different rules for being updated and influence each other in the perception of the world. This separation of new and old knowledge most likely exists in many layers rather than the two distinct categories represented here. The thought process is built of memory functions from the lowest sensory input to the highest forms of insight.This concept has been explored in the area of intelligent control as the principle of increasing intelligence with decreasing precision (Saridis, 1979).

The transition from memory-less neural networks to systems that incorporate the concepts of short-term memory closely compares to the transition between memory-less and memory capable digital circuits mentioned earlier. There are two common forms of memory circuits, and while both techniques make use of memory, one system creates outputs that are solely dictated by the current state of the machine while the second approach allows the output of a given state to be altered immediately by the inputs to the system (as well as by the current state). Systems that do not force outputs to be synchronized with their internal state tend more often to generate unstable outputs. These momentary instabilities as the system output changes before and after the internal state can cause a race in the circuit. Instead of proceeding to the correct next state, the system jumps to a state determined by the intermediate outputs.

Clearly, the human model makes use of the synchronized technique for dealing with inputs (as described earlier the integrating functions of the sensory system synchronize inputs to the frequency of brain activity). This simple restriction insures the stability of the output at the cost of a more complex minimum circuit. Since the brain is made up of a large number of simple elements, the trade-off seems perfectly equatable. Thus the model of the human neural system should include memory elements to allow it to handle the concept of time, and a closer look should be given to the hierarchic structure of the thought process. This should provide a wide variety of solutions for many temporal problems, but it must be noted that other features of the human brain have been simplified here. Most notably, the fact that signals traversing the brain do not travel instantaneously and the fact that there are delays in the transmission of impulses from one neuron to the next could well account for some of the brains ability to compute time-based functions.

Once a network is capable of processing time dependent stimuli, the next logical step is to attempt to develop some form of look- ahead or precognition. The anticipation of events that may occur based on experience. For this transition, the system must be capable of applying a learned set of rules to multiple scenarios. Each application of the learned rule base would create additional possible futures. This process could be sequential (a recursive approach) or in parallel. From observation of human cognition, these two options are both implemented on a daily basis. For the more common occurrences, the parallel technique is subconsciously employed . The expected outcome of many situations is known and the necessary response is supplied automatically from associations. In other cases, a set of rules must be applied one step at a time in order to determine some future condition. The choice of process is usually determined by the level of familiarity one has with the task at hand. For instance, in a chess game a novice player will ponder each move and counter move where an expert player may simply attempt to visualize the possibilities. Each of these two methods has advantages: the parallel system would respond rapidly to a complex scenario while the sequential process would not. On the other hand, the sequential system has a greatly reduced learning time.

Either of these two techniques could provide interesting results when attempting to model time dependent response to dynamic systems. In order to implement a given approach the existing constraints of the neural network system must be considered. The back-propagation algorithm does not support the feedback loops that would be required to implement a recursive solution to a temporal problem. That is, there is no facility to route the solution or neural output at time **N** back into the same network in order to project a solution for time **N+1**. This limitation encourages experimentation towards the parallel approach. A gang of neural responses are grouped to accept data in a format which will be used to imply time passage. The outputs of previous layers will be passed into the input of an identical layer (representing the system at time **N+1**). Inputs to the system will be presented as a stream of sequential samples of time dependent data. Inputs for time **N** are given to the first layer. Inputs for time **N+1** are sent to the second layer and so on. In this manner a series of actions may be represented in the neural model as a large static problem where the flow of input signals is known for a brief span of time. Data presented in this manner may represent a history of data as well as the future inputs to the system. In the history mode, the weighting functions learned would represent the use of the previous states of the system in determining the proper course of action. While in the future data mode a system could be trained to expect a chain of events. Combination of the two techniques would provide a system that could respond correctly to sequences of stimuli.

III. Conclusions and Future Plans

This project has always recognized an eventual need to experiment on a large scale parallel architecture and a parallel machine was indicated in the original proposal as a specification task in the final year of the research. An opportunity occurred last summer when the Computer Engineering Department was offered an NCube parallel supercomputer at cost. NCube was interested in placing a machine in Florida and specifically at UCF. The MBR project elected to support roughly two-thirdsof the funding for the machine. The balance of funds was paid by the Office of Naval Research and the Computer Engineering Department. In September of 1990, Martin Marietta Orlando Aerospace agreed to fund the conversion of the simANNs system (see discussion above) to run as a parallel program on the NCube and be given a graphical user interface. This activity will have a strong impact on the MBR project as it will allow us to accelerate our original parallel programming plans by one year. Justifications for the use of parallel hardware beyond what we have discussed in our reports and presentations may be found in (Denning & Tichy, 1990)

At present we are continuing to expand and improve on the neural-based associative store and are assisting, when necessary, with the NCube port of our simANNs system. Our main focus now is on determining the best mechanism for temporal processing while maintaining our domain of launch processing.

References

Carpenter, G.A., Cohen, M.A. and Grossberg, S. (1987) Computing with Neural Networks, *Science*, **235**, 1226-1229.

Denning, P.J. and Tichy W.F. (1990) Highly Parallel Computation, *Science*, **250**, 1217-1222.

Dreyfus, H.L. (1979) *What Computers Can't Do*, New York: Harper and Row.

Dreyfus, H.L. and Dreyfus, S.E. (1988) Making a mind versus modeling the brain: AI back at a branchpoint, *Daedalus*, **117:1**,15-43.

Grossberg S. (1988) Nonlinear Neural Networks: Principles, Mechanisms, and Architectures, *Neural Networks*, **1**, 17-61.

Grossberg, S. and Schmajuk, N.A. (1989) Neural Dynamics of Adaptive Timing and Temporal Discrimination During Associative Learning, *Neural Networks*, **2**, 79-102.

Heileman, G.L., (1989) An Object-Oriented Approach to the Simulation of Artificial Neural Networks, Ph.D. Dissertation, University of Central Florida.

Hopfield, J.J. (1982) Neural networks and systems with emergent collective computational abilities, *Proc. Natl'l. Acad. Sci.*, **79**, 2554-2558.

Hopfield, J.J. and Tank, D.W. (1986) Computing with Neural Circuits: A Model, *Science*, **233**, 625-633.

Kanigel, R. (1987) Learning at the Sub-Neural Level, *Mosaic*, **18:3**, 16-25.

Kohonen T. (1988) An Introduction to Neural Computing, *Neural Networks*, **1**, 3-16.

Lenat, D.J., Borning, A., McDonald, D., Taylor, C. and Weyer, S. (1983) KNOESPHERE: Building Expert Systems with Encyclopedic Knowledge, **IJCAI**, 167-169.

Minsky, M. (1985) *The Society of Mind*, New York: Simon and Schuster.

Myler, H.R. (1991) A Neural Model for Memory-Based Reasoning in Advanced Launch Operations, *Journal of Neural Network Computing*, at press.

Saridis, G.N. (1979) Toward the realization of intelligent controls, *Proc. of the IEEE*, **67**.

Schiffren, R.M. and Atkinson, R.C.(1969) Storage and Retreival Processes in Long Term Memory, *Psychological Review*, **76**, 179-193.

Stanfill, C. and Waltz, D. (1986) Toward Memory-Based Reasoning, *Comm. of the ACM*, **29:12**, 1213-1228.

Takeda, M. and Goodman J.W. (1986) Neural Networks for Computation: Number Representations and programming complexity, *Applied Optics*, **25:18**, 3033-3046.

Waldrop, M.M. (1985) Machinations of Thought, *Science*, **85**, 37-52.

Appendix A Results of Multiple Training Runs of the Initial Topology

Back-Prop (learning rate 4)

number of patterns trained 1 Tolerance .003 .009 .019
number of training passes 14 10 7
number of patterns trained 2 Tolerance .003 .009 .019
number of training passes 224 18 10
number of patterns trained 3 Tolerance .003 .009 .019
number of training passes 681 315 249
number of patterns trained 4 Tolerance .003 .009 .019
number of training passes 760 396 324
number of patterns trained 5 Tolerance .003 .009 .019
number of training passes 1900 385
number of patterns trained 6 Tolerance .003 .009 .019
number of training passes 2490 852 684
number of patterns trained 7 Tolerance .003 .009 .019
number of training passes 4788 1491 791
number of patterns trained 8 Tolerance .003 .009 .019
number of training passes 4064 2024 1104

Delta-Bar-Delta (learning rate 4)

number of patterns trained 1 Tolerance .003 .009 .019
number of training passes 14 9 7
number of patterns trained 2 Tolerance .003 .009 .019
number of training passes 180 12 6
number of patterns trained 3 Tolerance .003 .009 .019
number of training passes 534 336 279
number of patterns trained 4 Tolerance .003 .009 .019
number of training passes 644 452 380
number of patterns trained 5 Tolerance .003 .009 .019
number of training passes 1180 535
number of patterns trained 6 Tolerance .003 .009 .019
number of training passes 1218 882 708
number of patterns trained 7 Tolerance .003 .009 .019
number of training passes 2751 1596 1043
number of patterns trained 8 Tolerance .003 .009 .019
number of training passes 2968 1240

Appendix B Subnetwork Training Results

SINGLE LAYER "DICTIONARY" SUBNETS:

PART1B.NET Scheduler sub-layer network [1,5,1,3]

Number of patterns trained	3
convergence	.050
number of training passes	1138

PART2B.NET Weather sub-layer network [1,5,8,6]

Number of patterns trained	6
convergence	.050
number of training passes	1410

PART3B.NET Diagnostics sub-layer network [1,5,2,3]

Number of patterns trained	3
convergence	.050
number of training passes	489

PARTB.NET   Combines Parts 1B,2B,3B into one main net [2,15,11,1,9]

******* Simulation Result *******
Algorithm: Back-prop
Convergence Tolerance: 0.050
Learning Rates ---
Layer 1: 0.00
Layer 2: 4.00
Number of passes:
Layer 1 ---
Forward: 7047
Backward: 1994
Layer 2 ---
Forward: 7047
Backward: 1994
Reached solution: Yes

PARTC.NET   Results in binary pattern, representing output [3,15,11,1,3]
******* Simulation Result *******
Algorithm: Back-prop
Convergence Tolerance: 0.050
Learning Rates ---
Layer 1: 0.00
Layer 2: 0.00
Layer 3: 4.00
Number of passes:
Layer 1 ---
Forward: 100008
Backward: 100008
Layer 2 ---
Forward: 100008
Backward: 100008
Layer 3 ---
Forward: 100008
Backward: 100008
Reached solution: No   (note: did not converge)

PARTD.NET   Binary representation of individual output [2,15,11,3,9]

******* Simulation Result *******
Algorithm: Back-prop
Convergence Tolerance: 0.050
Learning Rates ---
Layer 1: 0.00
Layer 2: 4.00
Number of passes:
Layer 1 ---
Forward: 10917
Backward: 3920
Layer 2 ---
Forward: 10917
Backward: 3920
Reached solution: Yes

PARTE.NET   Float value representation of output pattern [3,15,11,3,5,9]

******* Simulation Result *******
Algorithm: Back-prop
Convergence Tolerance: 0.009
Learning Rates ---
Layer 1: 0.00
Layer 2: 0.00
Layer 3: 4.00
Number of passes:
Layer 1 ---
Forward: 27
Backward: 2
Layer 2 ---
Forward: 27
Backward: 2
Layer 3 ---
Forward: 27
Backward: 2
Reached solution: Yes

PARTF.NET   ASCII 5 bit representation of result[3,15,11,3,5,9]

******* Simulation Result *******
Algorithm: Back-prop
Convergence Tolerance: 0.100
Learning Rates ---
Layer 1: 0.00
Layer 2: 0.00
Layer 3: 4.00
Number of passes:
Layer 1 ---
Forward: 72
Backward: 53
Layer 2 ---
Forward: 72
Backward: 53
Layer 3 ---
Forward: 72
Backward: 53
Reached solution: Yes

S4-61
333524
P-118

Natural Language Knowledge Acquisition Systems

N91-70702

End of Year Report

Main PI: Fernando Gomez
Dept. of Computer Science
University of Central Florida
Orlando, Fl 32816
Jan-25-91

Table of Contents

Introduction

The purpose of our task is to automate the acquisition of knowledge from human experts. Our research has proceeded in two ways: (a) advancing the state of the art in knowledge acquisition using natural language and (b) transfering this technology to the actual construction of knowledge acquisition interfaces for real expert systems. As part of our transfer of technology, we are already building a prototype knowledge acquisition interface for OPERA. The interface written in Common Lisp is running in a Symbolics 3653, where we are making use of all graphic features of the Symbolics environment. This report centers in describing the main theoretical components of the knowledge acquisition interface to OPERA, its main milestones and the current effort. In the appendix, the reader may find copies of the papers which have been published during last year on the theoretical aspects of our research.

The goal of OPERA (Expert System Analyst) is to improve the operations support of the computer network in the space shuttle launch processing system. The checkout, control and monitor subsystem (CCMS) is a distributed computer network, which integrates software, microcode, display switches and hardware interface devices. OPERA is intended to function as a consultant to the operations staff assigned to each CCMS task. Two basic expert systems form OPERA: the Real Time System Error Manager (RTSEM) and the Problem Impact Analyst (PIA). When an error occurs, RTSEM displays information on this error obtained from a data base of errors. This information, although based on the CCMS message catalog information, contains experiential knowledge which "resides in the head of the human experts not in texts." The knowledge acquisition bottleneck that the designers of OPERA are presently experiencing is to gather this knowledge from the human experts and transfer it to OPERA in a form assimilable by the data structures and algorithms of the expert system. OPERA contains about one hundred thirty of these errors, but the actual number of errors in the computer network is beyond one thousand. Hence, OPERA is short in its knowledge base by a factor of ten.

The goal of our project is to build a knowledge acquisition interface by means of which a domain expert without knowledge of OPERA or expert systems will be able to transfer his/her

knowledge about the computer network errors to OPERA. One of the goals of this project is to take the English input as entered by an expert and transform it into the data structure which OPERA understands. But there are other goals in our project like helping the expert to organize his/her knowledge, refreshing his/her memory by bringing into place relevant pieces of information that he/she may have forgotten, and analyzing the precision and ambiguity of the English used by him/her to describe errors. (This is a task that we will start this coming year.)

Organization of Knowledge in an Expert Hierarchy

The first task which we have accomplished is to provide a way for the domain expert to organize his/her experiential knowledge. There has been a lot of evidence accumulated showing that human experts organize their knowledge into a hierarchy of concepts. This fact became evident to us during our initial interactions with OPERA experts. Experts use hierarchies all the time to describe the errors of the network system. They use these hierarchies all the time when they analyze errors and determine their causes and corrective steps. But for experts in this domain, the hierarchy is as natural as a botanical hierarchy is for a botanist. The hierarchy is the expert's underlying frame of reference for his/her domain knowledge. *They do not have to learn it.* We have built an initial hierarchy into our knowledge acquisition interface. This hierarchy contains the experiential and heuristic knowledge of the expert, and changes from expert to expert, making true the phrase that each expert has his/her own book. Accordingly, the interface allows different experts to build their own hierarchy. This hierachy contains the following information:

Knowledge specific to each error message, including:

* operational advisories
* diagnostic advisories
* probable causes

Given this framework, the interaction between the expert and the interface proceeds as follows:

(1) The expert is asked to select an error from a pre-stored table of errors.

(2) Then, the expert is asked to classify the error in the hierarchy.

(3) If the expert decides that the hierarchy is not adequately organized, he/she may decide to restructure it to meet his/her criteria. The Interface keeps multiple hierarchies associated with different experts.

(4) Once the expert has classified the error in the hierarchy, he/she is asked to enter information about that error, as explained in the section below.

All this interaction takes place by the expert touching with the mouse the nodes in the hierarchy. Note how the expert enters the information by typing English. This input is mapped into a frame structure and from there a program transform it into the OPERA data structures.

System Knowledge Organized into a Static Hierarchy

In order to further achieve our goal of helping the expert transfer his/her knowledge to the expert system, we have built a hierarchy which contains structural knowledge about the systems and devices. We have called this hierarchy the *static hierarchy*, because the knowledge in it can not be changed by the expert. The domain expert does not know that this hierarchy even exists. The interface uses this hierachy to help the expert enter information. For instance, suppose that a domain expert is ready to enter information about the diagnostic advisory of a specific error, then, automatically, the interface displays in one of the windows all the diagnostic categories. The expert can then click with the mouse in one of the categories, and the interface will display all subcategories within that category. Now, the expert may decide to choose one of those categories by clicking with the mouse. If he does so, that category will be automatically inserted in the text he/she is typing.

We can establish an analogy between these two types of hierarchies, which we have called *experiential* and *static*, and the hierachies of a diagnostician and a physiologist, respectively. Diagnosticians are trouble shooters. They organize their knowledge on the basis of experiences and cases. Physiologists have a deep knowledge about the organs and functions of the human body from a structural and causal point of view. Of course, there are situations in which a

diagnostician would like to know more about the function of organs, and they need to consult books during the diagnosis of a case. This is precisely the aspect which we are capturing in our interface. The knowledge in the static hierarchy is always ready to help the domain expert by bringing up relevant structural information about devices and programs.

The implementation of these two hierarchies has been done by constructing a unified data structure for all information. This structure facilitates ease of retrieval and updating of data, avoids redundant data and supports multiple access paths to information.

Current Effort

Our work will concentrate on the following areas: the testing of the present interface with domain experts and the analysis of the language used by domain experts. As result of this testing, the interface will require some changes. We do not anticipate major changes, but only small additions and refinements to the software components which are already implemented. In parallel with the testing process, we will be working on analyzing the English language entered by domain experts. This analysis will focus on two aspects: (a) uniformity of the language and (b) ambiguity. Presently, diverse experts use very different expressions to describe identical advisories and causes. In some cases, this does not pose any problem, but there are cases in which it is not clear that they are referring to the same situation. Consequently, we are going to design algorithms to uniformize as much as possible the language used by domain experts in describing the errors. Another aspect of our investigation will be to analyze the experts' language looking for ambiguity in the sentences. Our goal will be to investigate *lexical ambiguity* and the ambiguity of complex nouns. Syntactical and semantic ambiguity will not be part of our study during the first the quarter, although we may look into it during the second quarter, depending on our progress in analyzing lexical ambiguity. After this analysis is completed, our goal will be to help the expert define the terms as precisely as possible by pointing out ambiguous expressions and terms. The progress in this area, of course, will depend on the testing of the interface. If the current software does not require major modifications we plan to start

implementing some of the ambiguity algorithms during the first quarter of the second year.

Appendix A

Natural Language Knowledge Acquisition Systems

UCF Dr. Fernando Gomez
Richard D. Hull
Clark R. Karr

Grumman R. Bruce Hosken
William Verhagen

Presentation Agenda

- Introduction
- Plan and Schedule
- Summary of Work through last Review
- Current Effort
- Remaining Tasks

Introduction

Purpose and Task

- Purpose - To capture human expertise.

- Task - To create a natural language Interface whereby domain experts can transfer their knowledge to first generation expert systems.

Introduction

Typical Error Seen By Domain Expert

FEP 141 ($$$$) MICROCODE DID NOT RECEIVE AN ACKNOWLEDGE SIGNAL FROM THE I/O ADAPTER, DATA ACQUISITION HAS BEEN INHIBITED MICAS=$$$$, NSB=$$$$
* TERMINAL ERROR FOR THE GSE FEP. THE I/O ADAPTER DID NOT SNED AN ACKNOWLEDGE SIGNAL
TO THE MICORCODE DURING THE OPERATION INDICATED BY MICAS.

Probable Cause(s):
1. I/O Adapter failed.
2. GSE Option Plane failed.
3. I/O Adapter port on the 4-port controller failed.

Operations Advisory:
1. Halt the CPU, and record CPU registers. Push CPU through recovery.
2. If redundant FEP hasn't taken over, configure another FEP, or $CLAI existing FEP again.
3. $SPRCVE.
4. If redundancy isn't available, and original FEP fails to $CLAI, then troubleshoot per following diagnostic advisory.
5. Lookup the MICAS in the microcode listings, and verify the operation being executed at the time of the anomaly.

Diagnostic Advisory:
1. $DPLORT LI 5.
2. SEQ FEPID1, if errors occur I/O Adapter thumbin may assist troubleshooting.
3. GSE M02.
4. SEQ FEVTR1 (loop T/R via RCVS).

Introduction

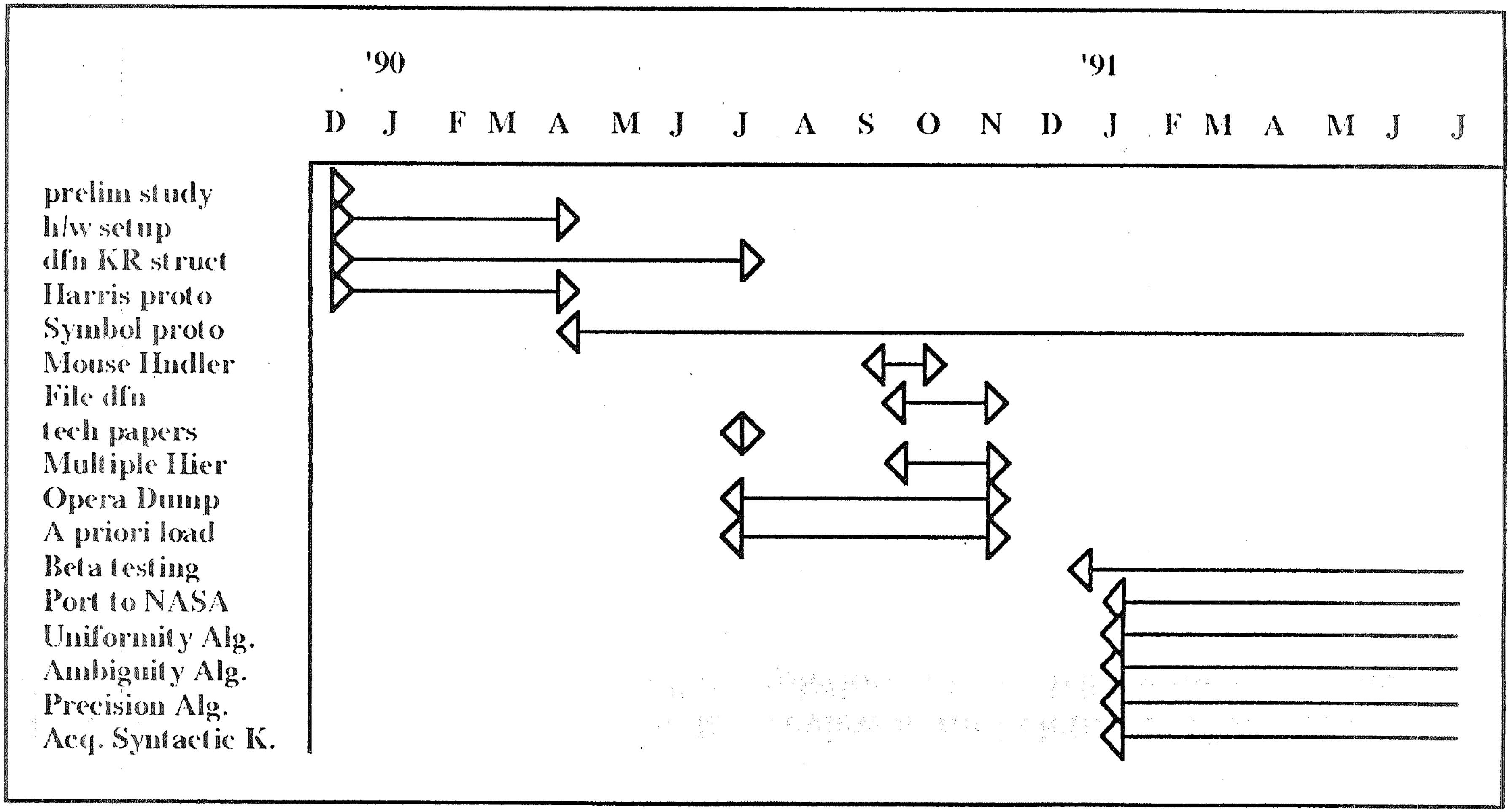
OPERA Representation of Message Data

** MSG-CAUSES: **
((FILTERS) 1. FEP I/O Adapter failed.)
((FILTERS) 2. GSE Option Plane failed.)
((FILTERS) 3. I/O Adapter on 4-port controller failed.)
•
•
** OPS-ADVISORY: **
((FILTERS) 1. Halt the CPU, and record CPU registers. Push CPU through recovery.)
((FILTERS) 2. If redundant FEP hasn't taken over, configure another FEP...)
•
•
** DIAGNOSTIC-ADVISORY: **
((FILTERS) 1. $DPLORT LI 5.)
((FILTERS) 2. SEQ FEPID1, if errors occur I/O Adapter thumbin may assist...)
•
•
(Message Text and Other Information used by OPERA in maintaining the Error Message Knowledge Base.)

C-3

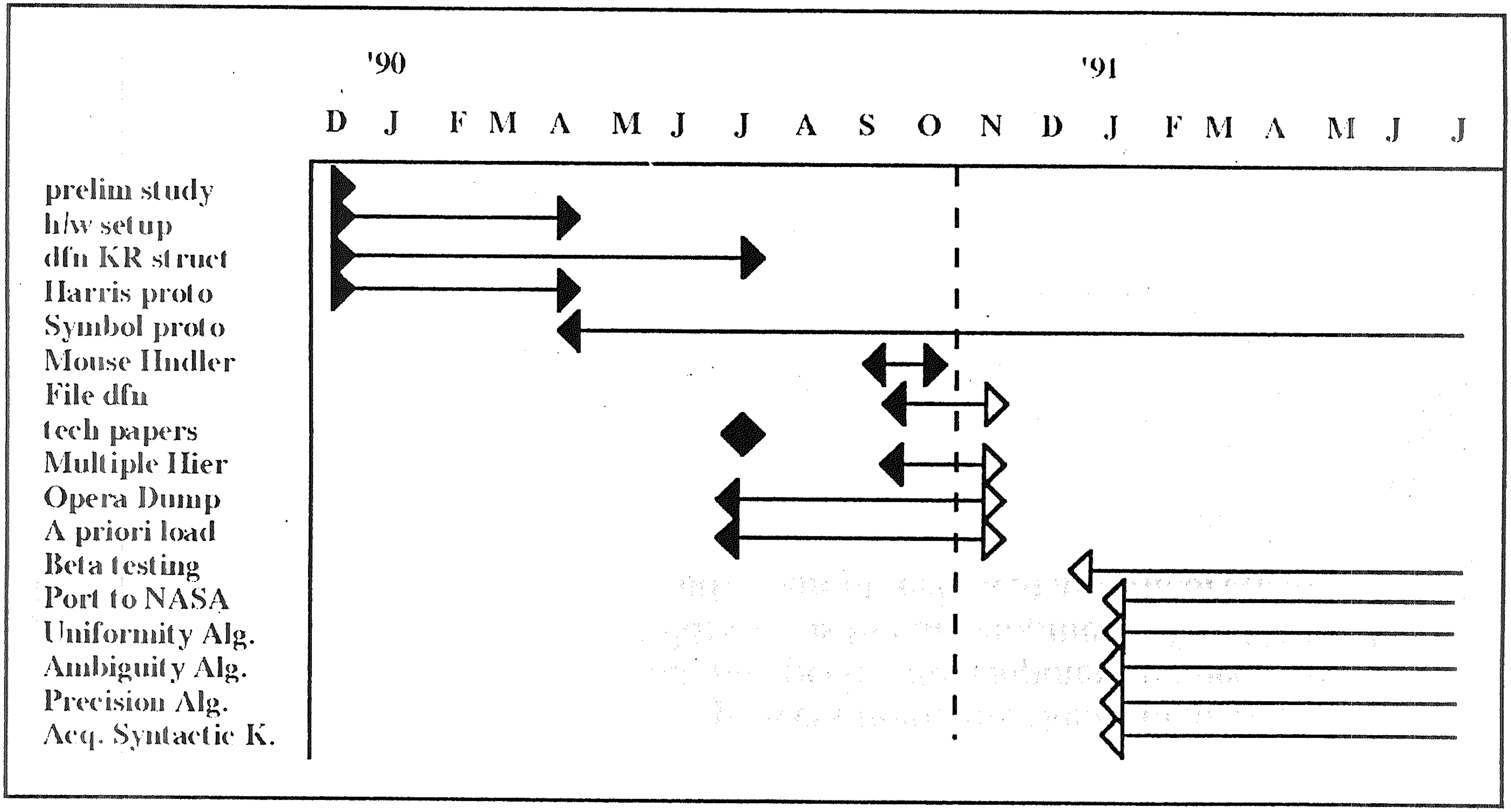
Plan and Schedule

Schedule



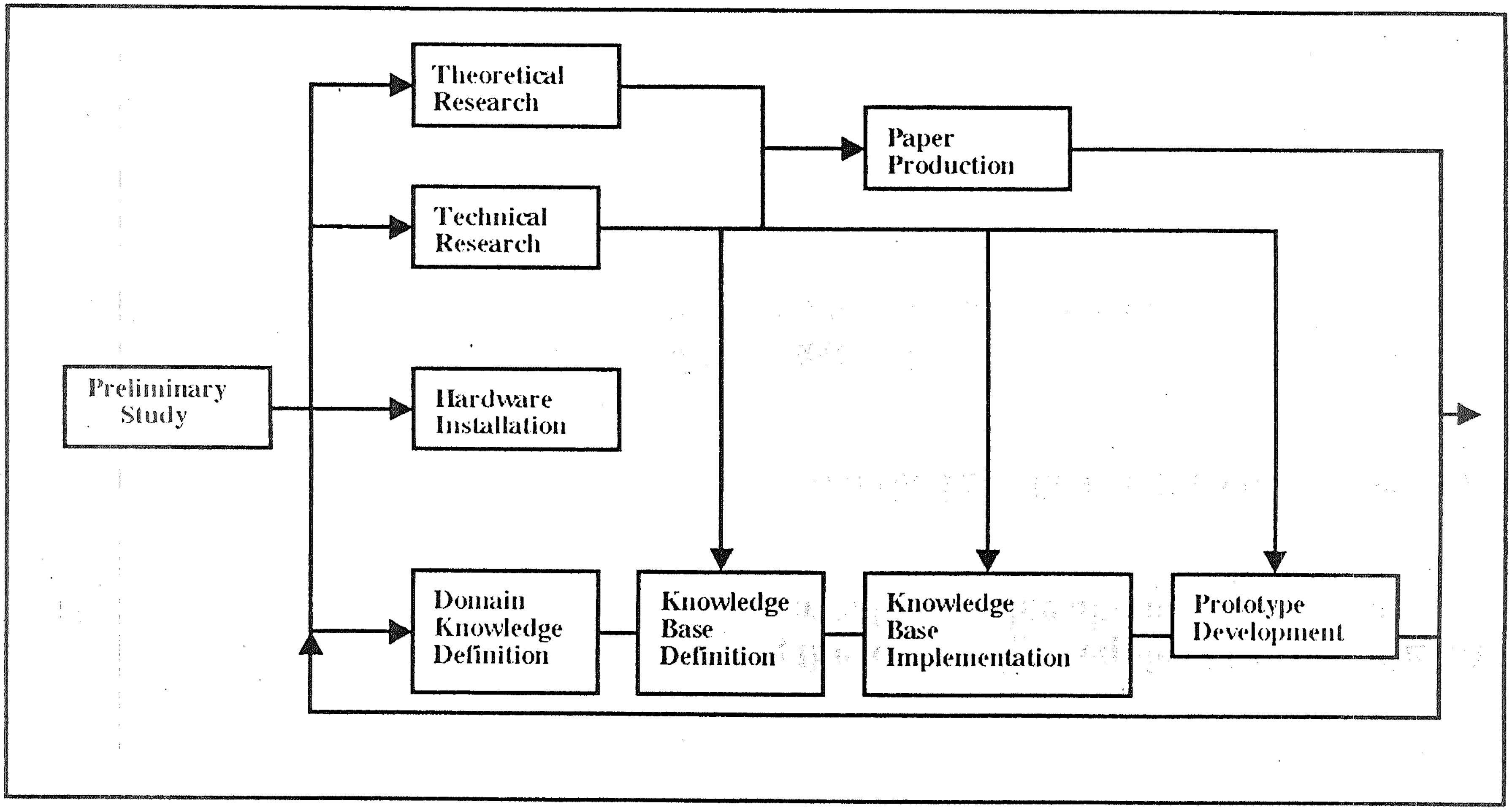
Plan and Schedule

Schedule as of Last Review



Plan and Schedule

Plan



Current Effort
Specific Tasks

- File definitions for a priori knowledge.
- Routines needed to dump data acquired by the interface in OPERA format.
- Multiple hierarchies (static and expert) existing within a common framework.
- Data load program for the knowledge engineer.
- Alpha testing of interface system.

Current Effort

Static Hierarchy

- Contains a priori information
  - operational advisories
  - diagnostic advisories
  - probable causes

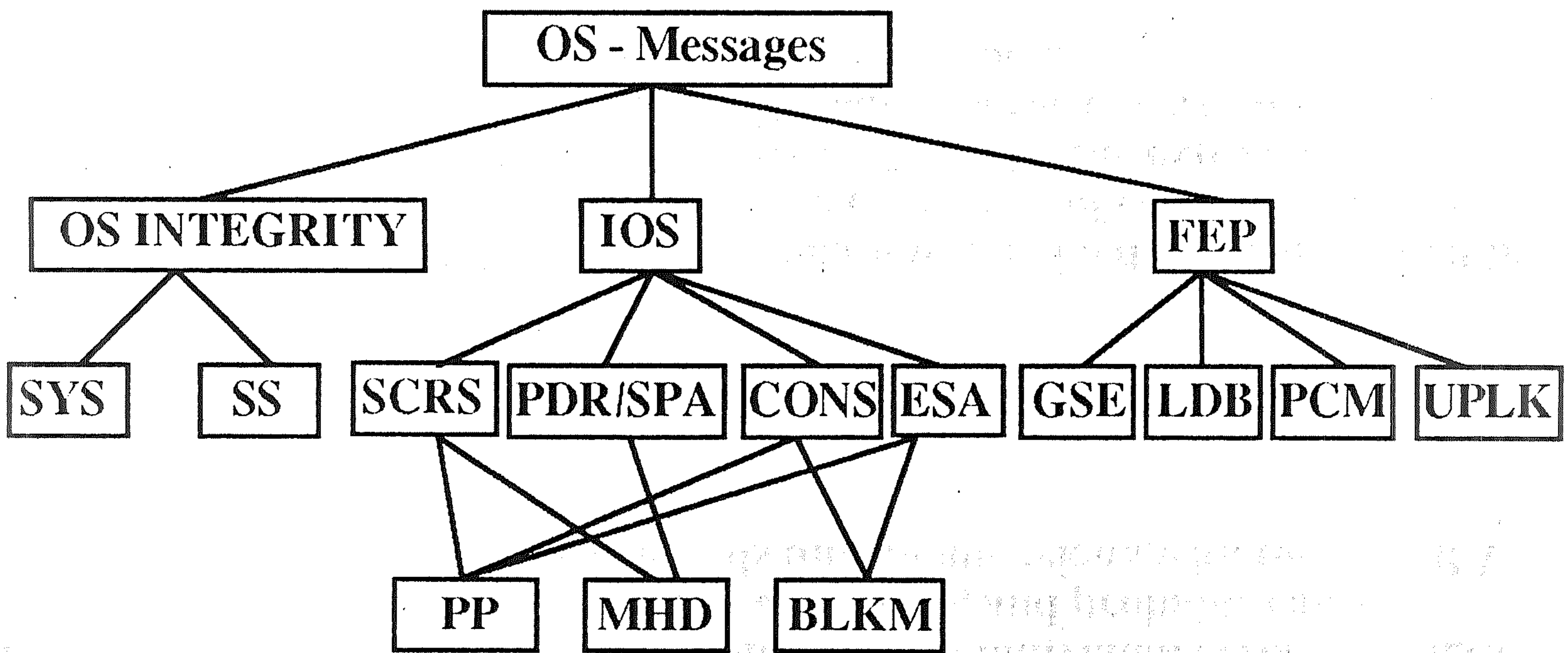
- Information visible to expert for selection during interview process.

- Not changable by domain expert

- Organization specified by Knowledge Engineer

Current Effort

Static hierarchy



Current Effort

Expert Hierarchy

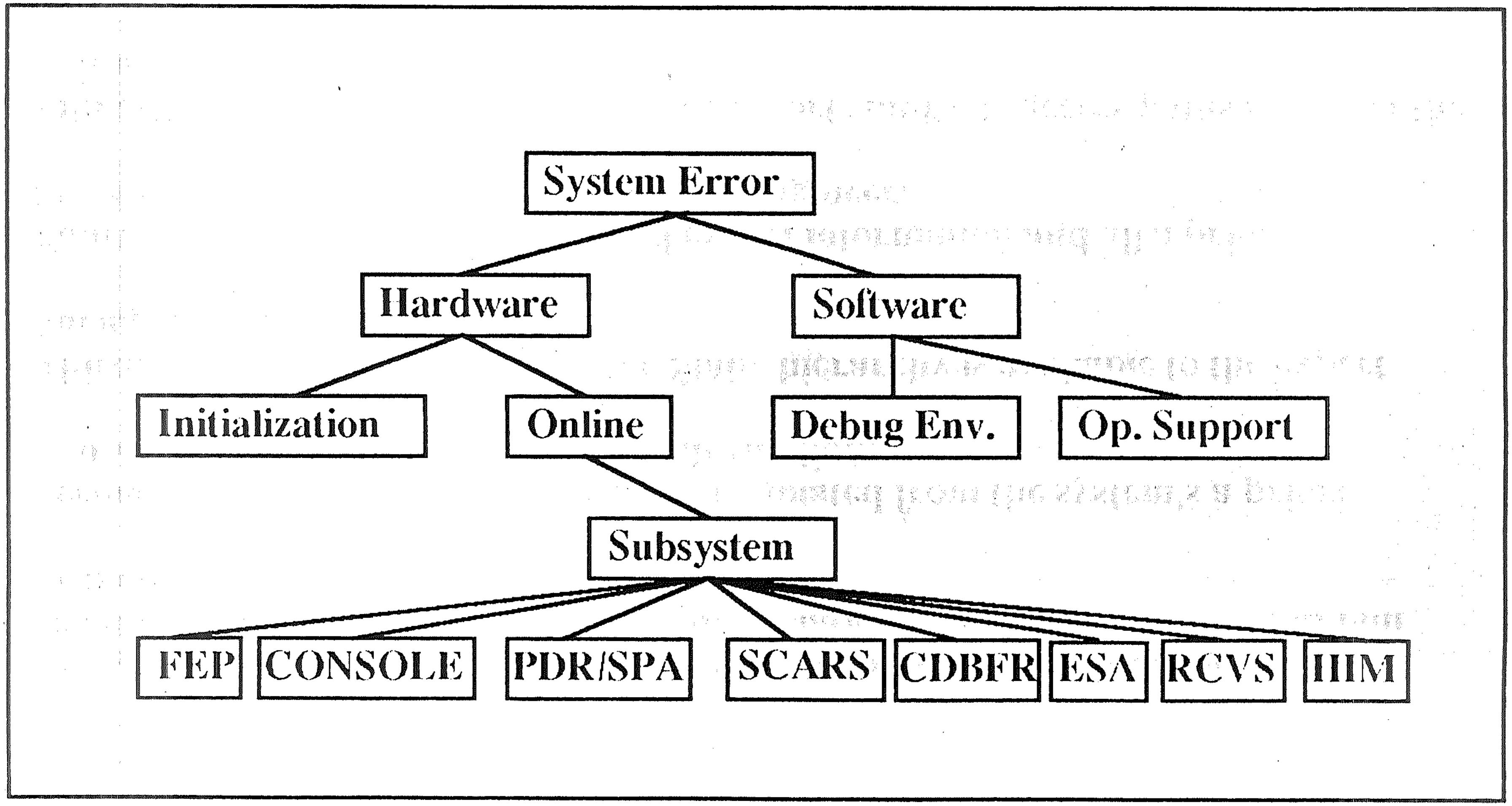
- Contains expert information
  - Knowledge specific to error message
    - operational advisories
    - diagnostic advisories
    - probable causes

- Selected information obtained from Static hierarchy.

- Changable by domain expert
  - Organization of hierarchy
  - Contents of nodes within hierarchy

Current Effort

Expert Hierarchy

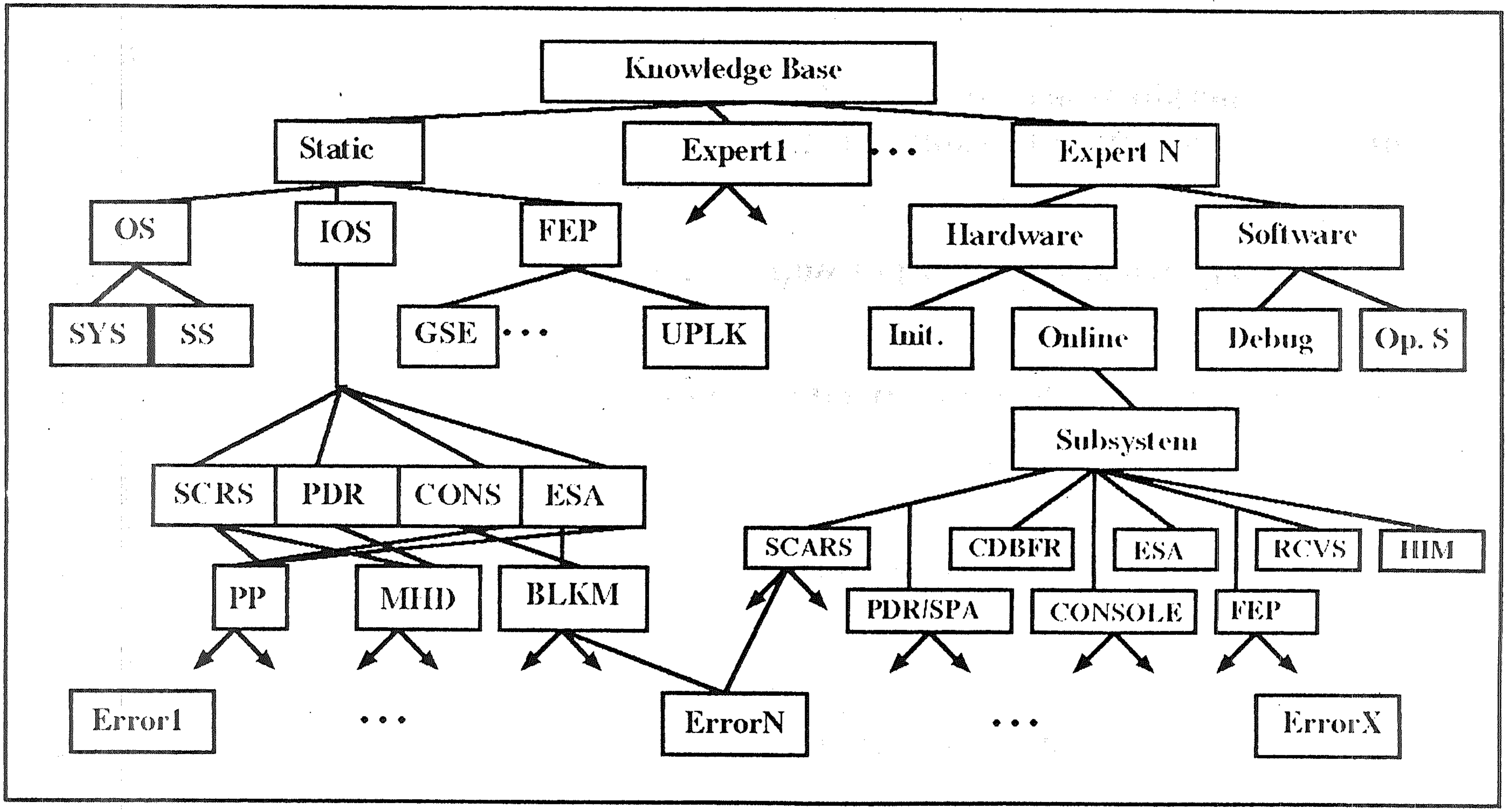


Current Effort
Underlying Knowledge Structure

- Single, unified data structure for all information.
    - Ease of retrieval and update of data.
    - Non-redundant data.
- Isolation of individual expert's knowledge.
- Static knowledge available to expert.
- All information available to Knowledge Engineer.
- Support multiple access paths to information.

Current Effort

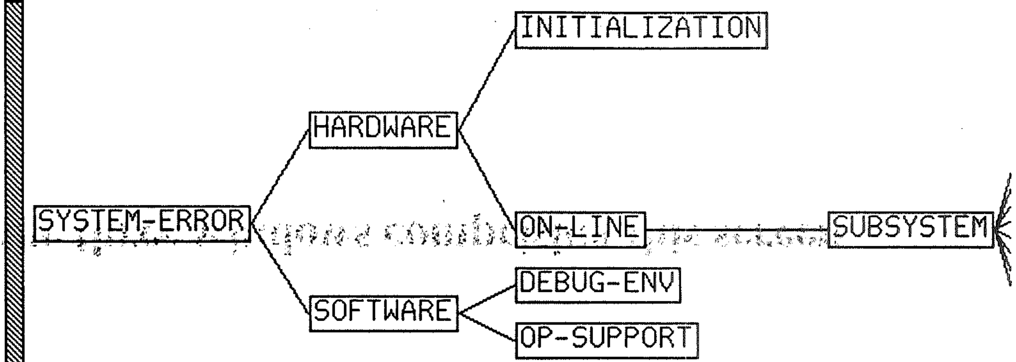
Underlying Knowledge Structure



Opera Knowledge Acquisition Interface

Clear Screen	Decode Status Regs	Delete Message	Describe Msg	Edit Message	Generate Dump
Help	Load Hierarchy	Preload Message	Restructure Hierarchy	Save Hierarchy	Show Hierarchy
Show Message	Static Hierarchy	UnDelete Message			

Expert Last Name: hull
Expert First Name: r



ORIGINAL PAGE IS OF POOR QUALITY

Opera Knowledge Acquisition Interface command: Load Hierarchy
Opera Knowledge Acquisition Interface command:

UCF Dept. of Computer Science NASA Grant 11-28-204

Message to Describe: 135-1

NIL
MESSAGE: 135-1

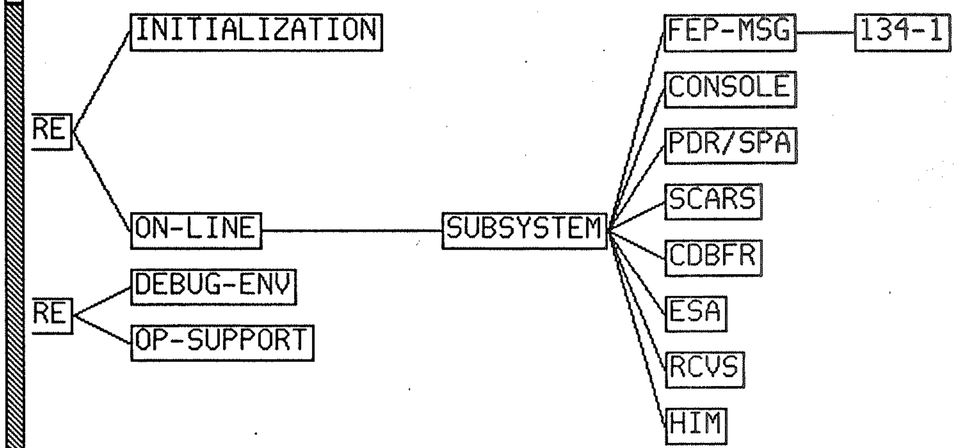
Is the category containing 135-1 in the hierarchy? (Y or N) Yes
.
What kind of message is 135-1? *FEP-MSG*

What type of message is 135-1
Enter Terminal or Non-Terminal: Terminal

Probable Causes:
1: HIM Control Card failed.
2: HIM Master Control Card failed.
3: Transmitter/Receiver in the HIM failed.
4: GSE FEP T/R failed.
5: Ground Data Bus problem.
6:

Operational Advisory:
1: *Gather HARD COPY output of $HWMON and $SYS pages for analysis.*
2: *Halt the CPU, record the CPU registers and perform a Recovery Dump.*
3: *Reload and reinitialize the failed FEP using $CLAI (Common Load and Initialization) program.*
4:

Diagnostic Advisory:
1:



Diagnostic Categories
SEQ FEPI, FEP I/O Adapter Diag
SEQ CDBK, Master Timing Unit Diagnostic
SEQ CP1, CPU DIAGNOSTIC PART 1
SEQ CP2, CPU DIAGNOSTIC PART 2
SEQ CP3, CPU DIAGNOSTIC PART 3
SEQ MEM, MEMORY DIAGNOSTIC
SEQ FPD, FLOATING POINT DIAGNOSTIC
SEQ SP1, SYSTEM PROTECT DIAGNOSTIC PART 1
SEQ SP2, SYSTEM PROTECT DIAGNOSTIC PART 2
SEQ OPD, OPTION PLANE DIAGNOSTIC
SEQ TIM, INTERVAL TIMER DIAGNOSTIC
SEQ MSM, 64K SEMI-CONDUCTOR MEMORY
SEQ CCPTST, CDBFR COMM. PROCESSOR TEST
SEQ CDBB, MICRO BUFFER DIAGNOSIC TESTS
SEQ CDBM, CDBF DIAGNOSTIC, HEX

Message to Describe: 135-1

NIL
MESSAGE: 135-1

Is the category containing 135-1 in the hierarchy? (Y or N) Yes

What kind of message is 135-1? *FEP-MSG*

What type of message is 135-1
Enter Terminal or Non-Terminal: Terminal

Probable Causes:
1: HIM Control Card failed.
2: HIM Master Control Card failed.
3: Transmitter/Receiver in the HIM failed.
4: GSE FEP T/R failed.
5: Ground Data Bus problem.
6:

Operational Advisory:
1: *Gather HARD COPY output of $HWMON and $SYS pages for analysis.*
2: *Halt the CPU, record the CPU registers and perform a Recovery Dump.*
3: *Reload and reinitialize the failed FEP using $CLAI (Common Load and Initialization) program.*
4:

Diagnostic Advisory:
1:

Diagnostic Categories

```
SEQ FEPI, FEP I/O Adapter Diag
SEQ CDBK, Master Timing Unit Diagnostic
SEQ CP1, CPU DIAGNOSTIC PART 1
SEQ CP2, CPU DIAGNOSTIC PART 2
SEQ CP3, CPU DIAGNOSTIC PART 3
SEQ MEM, MEMORY DIAGNOSTIC
SEQ FPD, FLOATING POINT DIAGNOSTIC
SEQ SP1, SYSTEM PROTECT DIAGNOSTIC PART 1
SEQ SP2, SYSTEM PROTECT DIAGNOSTIC PART 2
SEQ OPD, OPTION PLANE DIAGNOSTIC
SEQ TIM, INTERVAL TIMER DIAGNOSTIC
SEQ MSM, 64K SEMI-CONDUCTOR MEMORY
SEQ CCPTST, CDBFR COMM. PROCESSOR TEST
SEQ CDBB, MICRO BUFFER DIAGNOSIC TESTS
SEQ CDBM, CDBF DIAGNOSTIC, HEX
```

```
SEQ FEPI, FEP I/O Adapter Diag
TEST01 - Four-port Memory Checkout
TEST02 - Timer Checkout Using Self-test
TEST03 - Toggle I/O Register Checkout
TEST04 - Read/Write I/O Adapter Registers
```

Message to Describe: 135-1

NIL
MESSAGE: 135-1

Is the category containing 135-1 in the hierarchy? (Y or N) Yes
.
What kind of message is 135-1? *FEP-MSG*

What type of message is 135-1
Enter Terminal or Non-Terminal: Terminal

Probable Causes:
    1: HIM Control Card failed.
    2: HIM Master Control Card failed.
    3: Transmitter/Receiver in the HIM failed.
    4: GSE FEP T/R failed.
    5: Ground Data Bus problem.
    6:

Operational Advisory:
    1: *Gather HARD COPY output of $HWMON and $SYS pages for ana lysis.*
    2: *Halt the CPU, record the CPU registers and perform a Rec overy Dump.*
    3: *Reload and reinitialize the failed FEP using $CLAI (Comm on Load and Initialization) program.*
    4:

Diagnostic Advisory:
    1: Use *SEQ FEPI, FEP I/O Adapter Diag* and sub-test *TEST01 - Four-port Memory Checkout.*
    2: Also use *SEQ CDBK, Master Timing Unit Diagnostic.*
    3:

Predecessor Messages:
    1:

Remaining Tasks

- Continuing definition and implementation of knowledge base.

- Beta Testing.

- Port Interface to NASA.

- Natural Language
    - Ambiguity of Natual Language
    - Uniformity of Natural Language
    - Precision of Natural Language

Appendix B

4th AAAI-Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop
Banff, Canada, October, 1989.

Knowledge Acquisition From Natural Language For Expert Systems Based On Classification Problem-Solving Methods

Fernando Gomez
Department of Computer Science
University of Central Florida
Orlando, Fl 32816
gomez@ucf.csnet

Abstract

It is shown how certain kinds of domain independent expert systems based on classification problem-solving methods can be constructed directly from natural language descriptions by a human expert. The expert knowledge is not translated into production rules. Rather, it is mapped into conceptual structures which are integrated into long-term memory (LTM). The resulting system is one in which problem-solving, retrieval and memory organization are integrated processes. In other words, the same algorithm and knowledge representation structures are shared by these processes. As a result of this, the system can answer questions, solve problems or reorganize LTM.

1. Introduction

Can a simple expert system consisting of a hundred rules be designed from a natural language description by a human expert? Or can an expert system be updated from a natural language description of some of its components? In this paper, we present an integrated and domain independent system which builds classification-hierarchies from texts, retrieves information from them and solves problems.

We have been investigating the relation between comprehension and problem-solving. Our investigation has been guided by the idea that problem-solving and comprehension are not two separate processes each one with its own knowledge representation structures, but that they spring from a common source and share the same knowledge representation structures. It is beyond the scope of this paper to discuss these issues in detail, rather we center our discussion in providing evidence on how certain kinds of expert systems can be built from a natural language description by a human expert. We will show that the problem solving method used by the expert system to solve problems is the same as that used by the comprehender to keep memory organized and answer questions requiring chains of inferences.

We can roughly characterize the domain of the expert systems to which we have applied our method as consultation tasks. For instance, the tasks of building an expert system to select a programming language, or select a physician, etc. Some of these tasks are described in (Boose and Gaines, 1987). We can further characterize the scope of our system by the problem-solving method used to find solutions to a problem. The problem-solving method is that of classification problem-solving (Gomez and Chandrasekaran, 1984; Clancey, 1986). Classification-problem solving consists of conceptualizing the domain of the problem-solver into a hierarchy of concepts and designing an algorithm which solves a problem by placing it in the hierarchy of concepts. In (Gomez and Chandrasekaran, 1984) it is stated:

Given these principles of organization of medical knowledge, the solution of a medical

This research is supported by NASA-KSC Contract NAG-10-0058

case becomes a problem of taxonomic classification. It is similar to the problem of placing, say, a specimen of maple in a hierarchy of botanical concepts. It consists of identifying its superordinate and subordinate concepts.

The method to recognize problem solutions by placing them in a hierarchy of concepts was that of using clusters of production rules organized under the concepts in the hierarchy, called *specialists* (Gomez and Chandrasekaran, 1984). In this paper, (a) we will show how to automatically build the classification hierarchies from natural language, and (b) we will describe an algorithm which does not use production rules to place the problem solutions in the hierarchy.

Among the knowledge acquisition systems which have been developed, ETS (Boose, 1984) and AQUINAS (Boose and Bradshaw, 1987) are the closer to our approach in the sense that these systems are domain independent and are based on building classification hierarchies for the solution of the problems. However, the differences are remarkable since we are not using elicitation techniques to build the classification hierarchies, but we are building them from natural language descriptions. Also, we are not translating the expert's knowledge into production rules, but we are using an general algorithm to produce solutions from the constructed hierarchies. Nano-KLAUS (Haas and Hendrix, 1980) was an earlier natural language acquisition system. However, their system is not targeted to the construction of expert systems, and it does not deal with the issues which are essential to our approach. The differences from other knowledge acquisition systems, MORE (Kahn, Nowlan and McDermott, 1985), MOLE (Eshelman, Ehret, McDermott and Tan, 1987) and SALT (Marcus, 1987), are more striking since these systems do not use classification problem-solving and presuppose domain knowledge.

This paper is organized as follows. In part 1 consisting of sections 2 and 3, we briefly explain some of the ideas on which our method is based, and in part 2 consisting of sections 4, 5, 6, 7 we describe how these ideas are applied to the construction of expert systems.

2. Background

For the last few years, we have developed a model of comprehension of elementary scientific texts or expository texts and a program which embodies the model, called SNOWY, (Gomez, 1985; Gomez and Segami, 1989a, 1989b). The comprehender of our model is a reader who is unfamiliar with most of the concepts which he/she is reading about. When SNOWY starts reading a scientific paragraph its domain knowledge consists only of a bare hierarchy of concepts. As SNOWY reads, new concepts are integrated in the right place in the hierarchy and new relations about already known concepts are learned. For instance, after reading the passage below with no knowledge about hormones and insulin, SNOWY acquires the concepts *hormones, insulin,* and *level of sugar in the blood.*

Hormones are chemicals produced by animals. Hormones control the activities of cells. Insulin is a hormone produced by the pancreas. Insulin controls the level of sugar in the blood by making cells take in sugar.

SNOWY does not only acquire those concepts, but it also builds a classification hierarchy. For instance, the concept *insulin* is classified as a subconcept of *hormone produced by the pancreas;* this, as a subconcept of *hormones;* this, in turn, as a subconcept of *chemical*

produced by animals and, finally, this last one as a subconcept of *chemicals*. In contrast to knowledge-intensive models of comprehension which understand by framing the input in rich pre-built structures, SNOWY understands by using *formation* rules which build conceptual structures from the logical form of sentences. These conceptual structures are then passed to a *recognizer* algorithm which checks if those concepts exist in long-term memory (LTM). Finally, those concepts which the recognizer algorithm fails to recognize are passed to an integration algorithm which decides where to integrate those concepts in LTM. We have called these three phases concept formation, concept recognition and concept integration.

Although SNOWY is a knowledge-lenient model of comprehension, this does not mean that SNOWY does not possess any knowledge when it starts reading a text. Besides the linguistic knowledge which allows SNOWY to parse and disambiguate sentences (Gomez, 1988), SNOWY posseses a set of categories organized in a hierarchy, a set of primitive relations for action and descriptive verbs, which are used to build the logical form of the sentence, and a set of formation rules which build the LTM conceptual structures from the logical form of the sentences. Figure 1 contains a sample of the *a priori* hierarchy. The hierarchy is divided into physical things and ideas, as shown in Figure 1. The portion of the hierarchy which is shown in Figure 1 is strictly hierarchical. However, when the system begins to acquire new concepts, it builds a "tangled" hierarchy as described in the sections below.

As new sentences are typed, the system builds the representation of the new concepts and relations and inserts them into the hierarchy at the appropriate places. For example, for the sentence "germs are animates," the system adds the following to LTM:

(germ (is-a (animate)))
(animate (classes-of (germ)))

Before describing how to build these hierarchies, we need to explain briefly the knowledge representation structures used by our system. Three kinds of representation structures may be part of LTM: object-structures, action-structures, and event-structures. For purposes of this

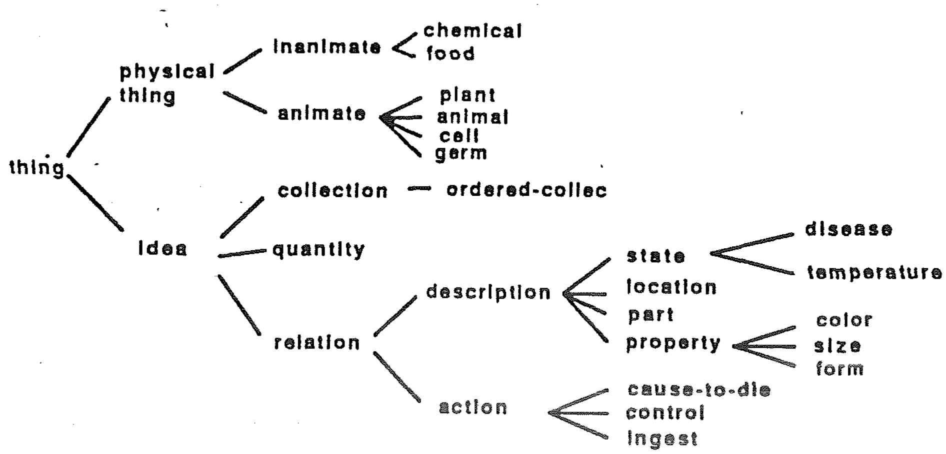


Figure 1: Partial content of the initial hierarchy of concepts in LTM

paper, we need only to describe object-structures and action-structures; event-structures are explained in (Gomez, 1986). All these structures are variants of the frame notation. An object-structure contains knowledge about physical or abstract objects and has slots called conceptual relations or predicates. Two kinds of conceptual relations can be distinguished: those which describe objects and those that attribute actions to physical things. This distinction corresponds to the distinction between descriptive and action verbs in natural language (Gomez, 1982). "Hormones are chemicals" and "all centipedes are slim and have many legs" are descriptive sentences, while "antibiotics kill germs" and "cells take in sugar" express actions performed by physical things. Descriptive attributes are also represented following the frame notation; for instance, the knowledge in the sentence about centipedes is represented as:

centipede
  (form (slim (q1 (all))))
  (has-body-part (leg (q1 (all)) (q2 (many))))

The slot q1 contains the quantifier of the concept described by the object-structure, and the slot q2 contains the quantifier of the second argument of the relation. Conceptual relations denoting actions are represented in the *object-structure* as:

relation (a1) (if the relation is monadic)

If the relation takes two arguments or more it is represented as:

(relation (concept1 a1))

where *concept1* is the second argument of the relation. If the same relation is true of two or more concepts, as is the case of the relation underlying the sentence "Penguins live in the sea and in the land," the relation is represented in the *object-structure* as:

(relation (concept1 a1) (concept2 a2) ... (concepti ai))

The terms *a1*, *a2*, ... *ai* stand for the names of the conceptual relations. The structure which represents the conceptual relation itself is called an *action-structure*. For example, the action-structure representing the conceptual relation in the sentence "antibiotics kill germs" looks like:

a1	**(inst a1 cause-to-die)**
(args (antibiotics) (germs))	**(actor a1 antibiotics)**
(pr (cause-to-die))	**(object a1 germs)**
(q1 (?))	**(quantifier-subject a1 unknown)**
(q2 (?))	**(quantifier-object a1 unknown)**

Of these two representations, we will use the one on the left side. The one on the right side is a slot-assertion notation (Charniak and McDermott, 86). The slot *args* contains the arguments of the relation. If the relation is monadic, the slot *args* will contain one concept. If the relation is diadic (as in this case), the slot *args* contains two concepts, and so on. The first concept in the slot *args* is always the actor of the action verb and the second concept is the object. The slot *pr* contains the relation. The slot *q1* contains the quantifier of the first argument, and the slot *q2* has the quantifier of the second argument of the relation. If the relation is triadic, there will be a *q3* slot for the quantifier of the third argument. The representation of "antibiotics kill germs" will be done by using the following object-structures and action-structure:

```
antibiotics                          germs
  (cause-to-die (germs a1))          (cause-to-die:by (antibiotics a1))

                     a1
                     (args (antibiotics)(germs))
                     (pr (cause-to-die))
                     (how (?))
                     (q1 (?))
                     (q2 (?))
```

Every concept which fills a case (actor, object, destination, source, etc.) in the sentence is indexed independently. In the above example, both the actor, *antibiotics*, and the object, *germs*, are indexed using an object-structure. Note how the concepts *germs* and *antibiotics* point to the same a1-structure, so that the question "what kills germs"? can be answered. The question mark in the slots *q1 and q2* means that the sentence does not specify the quantifiers. If the sentence had said "all antibiotics kill germs," then the content of the slot *q1* would have been *all*. If the sentence had said "most antibiotics kill germs," the slot *q1* would contain *most*. Figure 2 contains the representation of "the human body kills germs by producing antibodies."

3. The Detection and Construction of Classification Hierarchies

The importance of explicit classification has been widely recognized and described by cognitive scientists working in reading. In explicit classification, the classes are clearly introduced and defined as in:

> **There are two kinds of mammals: terrestrial mammals and aquatic mammals. Aquatic mammals include the carnivorous and the omnivorous. The carnivorous includes the seals and walruses. The omnivorous includes the whales and etc.**

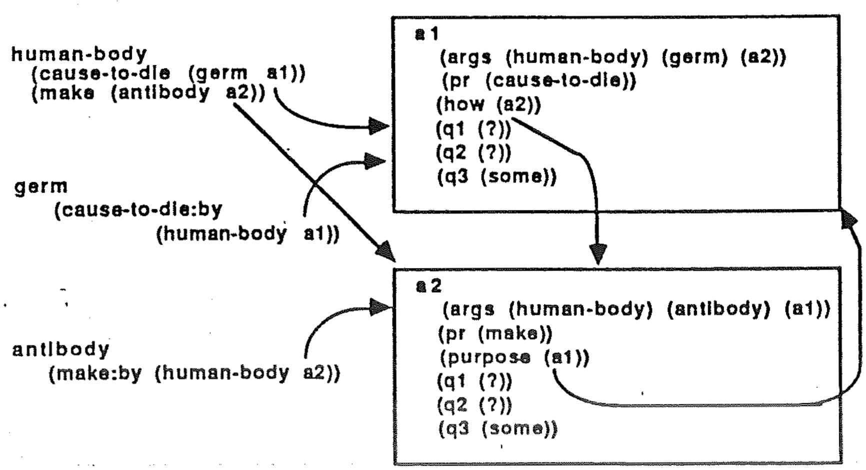


Figure 2: The representation of "the human body kills germs by producing antibodies"

In contrast to this type of classification, we will show that if a program is to acquire knowledge by reading texts it needs to classify in many instances in which classification is not apparent *prima facie*. Underlying classification is a more basic phenomenon than that which can be observed in other forms of classification, such as generalization-based classification or conceptual clustering. Furthermore, it is essential for constructing a problem solver from an English description of the task by a human expert, since the system will detect the classification hierarchies even when the expert does not see them. In the next sections, we will show that classification hierarchies underly two frequently used sentential structures: restrictive relative clauses and existentially quantified sentences.

3.1 Learning Concepts Denoted by Restrictive Relative Clauses

A first aspect of knowledge-acquisition is posed by the learning of concepts denoted by noun groups and restrictive relative clauses, e.g. "birds with long legs," "hormones produced by the pancreas," "programming languages which run in PCs," etc. The acquisition of these concepts involves their appropriate integration in the hierarchy, which in turn requires their classification by determining their parent and children nodes. For instance, consider the passage:

> **Birds with long legs live in swamps. They have a long bill to extract food from under water.**

This text introduces the concept *bird with long legs* for which a name is not given. However, a unique name in memory needs to be created for that concept so that additional knowledge about this concept can be learned by storing this knowledge under the same node, and so this node can be linked to the other nodes in the hierarchy. The acquisition of the concept *bird with long legs* entails its classification so that it can be integrated appropriately in the hierarchy of concepts. If the system fails to classify the concept, it will not be able to answer such questions as "are there birds with long bills?" or "do birds with long legs have feathers?"

Restrictive relative clauses and complex noun groups are represented in LTM via object-structures with a dummy name (a gensym), and their representation is characterized by the presence of a *characteristic-features* slot (written *cf*), whose purpose is to identify the concept by describing the features that are characteristic of the concept. For example, when building the representation of the sentence "insulin is a hormone produced by the pancreas," the concept *hormones produced by the pancreas*, a subconcept of the concept *hormones*, is created with the following representation:

(x1 (cf (is-a (hormone)) (make:by (pancreas (q2 (?))))))
(hormones (classes-of (x1)))

Within the *characteristic-features* slot there is always at least an is-a slot, which indicates the parent node of the concept being learned. In this example, the concept x1 has been created. X1 stands for a subconcept of the concept *hormones*, as indicated by the is-a slot, and is further characterized by referring to those hormones made by the pancreas. The acquisition of the concept x1 is completed by creating the slot classes-of in the concept *hormones*, and by filling it with x1. Had the slot classes-of existed in the concept *hormones*, then x1 would be added to it. The newly-created concept is identified by the content of the *characteristic-features* slot and not by the dummy name *x1*. Thus, if the question "what do hormones

produced by the pancreas cause?" is asked, the system recognizes that the concept *hormones produced by the pancreas* was previously created and accesses the object-structure x1. (The slot q2 contains "?" because it is unknown if every pancreas makes hormones.)

The acquisition of concepts described by complex noun groups is done in a similar manner. The representation of *bird with long legs* proceeds by creating the concept *long leg* first, and then the concept *bird with long legs.* Both concepts are represented in Figure 3. This definition creates the concept x1, a subconcept of the concept *leg*. The concept *bird with long legs* is represented by x2. The *characteristic-features* slot in x2 says that x2 is a subconcept of *bird* characterized by having long legs. Because the noun group does not specify how many long legs x1 has, the quantifier slot q2, whose scope is x1, has a question mark in it. If the system has knowledge about birds, it fills this slot; otherwise, it will remain with a question mark until the system finds out more about birds.

3.2 Formation or Learning Rules

The mechanism we have described for acquiring concepts denoted by restrictive relative clauses and noun groups is based on *formation or, learning rules.* These are very general rules anchored in the verbal primitives, and in some instances in lexical items. In most cases these rules use knowledge in LTM and in the structure built by the parser.

The formation rules which form the concepts x1 and x2, which have been explained above, are part of the formation of noun groups. For instance, the antecedent of the rule which is fired in the example above is:

> **The head noun has the feature animate and is followed by the preposition "with," and the object of the preposition has the feature body-part.**

Since the object of the preposition is also a noun group, the formation rules for the noun group are activated recursively to form the concept *long leg*. In this case, the antecedent of the rule activated is:

> **The head noun has the feature body-part and its modifier has the feature size.**

The latter rule will form the concept x1 and will pass it to the former rule, which will form the concept x2.

This has been a very simple example to illustrate basic formation rules. In the next section, we explain how formation or learning rules are used to learn more complex classification hierarchies.

x1	x2
(cf(is-a (leg))(size (long)))	(cf(is-a(bird)(body-part(x1 (q2(?)))))
leg	bird
classes-of(x1)	classes-of(x2)

Figure 3: The Representation of bird with long legs

3.3 The Acquisition of Classification Hierarchies

As we have seen, a classification hierarchy may be indicated explicitly, as in the sentence "antibiotics are chemicals" or it may be stated in a more subtle way as in the phrase "hormones produced by the pancreas." There is yet a more complex way of expressing a complex hierarchy of concepts like the illustrated in the passage below:

(1) Not all animals which live in the sea are fish. Some are mammals.

These two sentences introduce three concepts: *animals which live in the sea, sea fish* and *sea mammal.* The last two concepts are subconcepts of the first one.

The learning of these concepts and the hierarchy which they form is done by means of a learning rule attached to *not all* and by one of the learning rules anchored in *are.* The first sentence is parsed into:

(subj (not all animals) rela (g00001) verb (are) pred (fish))

g00001
(subj (animals) verb (live) prep (in the sea))

The learning rule which builds the representation for restrictive relative clauses is activated. This rule activates the formation rule for the verb *live.* The structure built is:

x1	**animal**
(cf(is-a(animal)(habitat (sea)))	**classes-of (x1)**

The name of the concept which has been created replaces the concept modified by the relative clause in the main sentence, resulting in:

(subj (not all x1) verb (are) pred (fish))

The formation of the main sentence proceeds by activating the learning rules for *are.* The following rule fires:

if the subject is under the scope of an existential quantifier or the negation of an universal quantifier then

a. create a concept with a dummy name, say x2, and with a characteristic-features slot containing the slots is-a (subj) and is-a(pred)

b. create another concept with a dummy name, say x3, and with a characteristic-features slot containing the slots is-a (subj) and is-a (?).

The terms *pred* and *subj* in the rules are variables containing the subject and the predicate of the sentence, respectively. This rule when applied to (1) gives:

x2	x3
(cf (is-a (x1)(fish)))	(cf (is-a (x1) (?)))

The application of the rule for restrictive relative clauses and the one for *are* has resulted in the creation of three concepts: the concept x1, which corresponds to *sea animal,* the concept x2, which corresponds to *sea fish,* and x3, which has been only partially specified.

Adverbs such as *almost, only,* and the the negation of universal quantifiers, e.g., *not all,* introduce links which connect the sentence where they appear to the next sentence. In order to solve these links, these lexical terms have attached to them formation rules. These rules are activated as part of the formation of the sentence which immediately follows the one which starts with *not all,* etc. These formation rules resolve the anaphoric reference of *some* in the second sentence and fill in the question mark in the is-a slot of the concept x3 above. The concepts and the hierarchy formed for the passage *Not all animals which live in the sea are fish. Some are mammals.* are found in Figure 4. The formation rules for *not all* do not only deal with this example, but also with examples such as:

Not all animals which live in the sea are fish. Whales are mammals.

In this case, the formation rule checks LTM to see if whales live in the sea. If so, it will replace the question mark in the concept x3 above with *mammal,* and create *whale* as a subconcept of x3. However, even if there is no knowledge about *whales* in LTM, the system infers from the structure of the sentence that *whales* are sea animals.

3.4 The Acquisition of Concepts Denoted by Existentially Quantified Sentences

The acquisition of concepts introduced by sentences starting with an existential quantifier requires the construction of a classification hierarchy in the same manner as we have done for relative clauses. The sentence "some mammals live in the sea" introduces the concept *mammal which live in the sea.* However, the representation in Figure 5 is inadequate for the job. The problem with this representation becomes obvious if the next sentence is "these mammals are intelligent." Since the structure above has not created the subconcept *mammal which live the sea* there is no node in the hierarchy to integrate the knowledge that they are intelligent. The representation of the sentence "some mammals live in the sea" is given below:

x1 (cf (is-a(mammal)) (habitat (sea))) --- is-a ---> mammal

X1 represents the concept *animals which live in the sea.* The concept *mammal* will contain

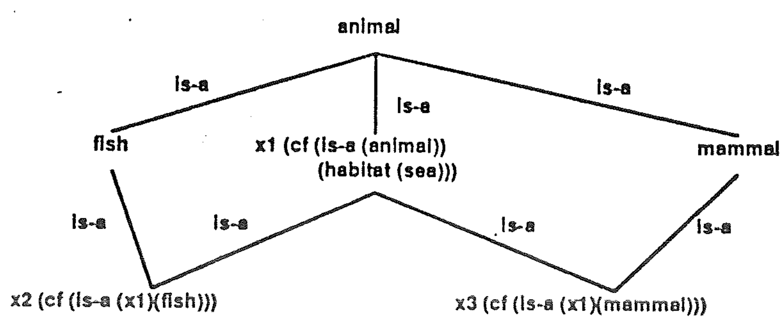


Figure 4: The hierarchy constructed for "Not all animals which live in the sea are fish. Some are mammals."

mammal   sea
habitat (sea a1)   habitat: by (mammal a1)

a1
args(mammal,sea)
relation (habitat)
q1 (some)
q2 (?)

Figure 5: Inadequate representation of "some mammals live in the sea."

the slot "classes-of (x1)." The knowledge expressed by the sentence "these mammals are intelligent" can now be integrated under the concept x1.

The role of classification is one of the most pervasive aspects in the structure of expository prose. As a final example consider the text below:

> Some birds are unusual because they cannot fly. The emu, the rea, and the ostrich are such birds. The emu is an Australian bird. The rea lives in South America. The ostrich lives in Africa and runs very fast.
>
> Some birds are unusual because they form a partnership with an animal. The oxpecker eats insects from the rhinoceros. The honey guide will lead a badger to a bees' hive.

The author found this text in a 6th grade comprehension exercise to teach children classification. The children were asked to write the names of birds on the dotted lines which follow the text. This text illustrates clearly that only through classification can the concepts and relations in the text be learned.

4. The Application of These Ideas to the Automatic Construction of Expert Systems

We will show in this section how these ideas can be applied to the construction of an expert system. The task which we will study is that of building a consultation system. We will use an example similar to that discussed in (Boose and Bradshaw, 1987), the selection of a programming language.

The expert is instructed to present his/her subject in a top down fashion describing first the most general concepts and then refining them until reaching the most concrete concepts which will form the tip nodes in the hierarchy. The paragraph below is just one way of how this

description may be accomplished. Many other are possible. After reading this paragraph, SNOWY builds the hierarchy of concepts shown in Figure 6.

> **There are several kinds of programming languages. Programming languages which manipulate symbols, called symbolic languages, are used for AI applications, while numeric languages are used for scientific applications. There are two main symbolic languages: Lisp and Prolog. Lisp is used for general AI applications, whereas Prolog is used when backtracking is needed. Common Lisp, Franz, Scheme and Golden Hill are Lisp languages. Franz runs in minicomputers and takes more than 3 megs of memory. Scheme is a good dialect of Lisp for PCs. It takes 0.8 meg of memory. Golden-Hill is also a version of Lisp for microcomputers, but it requires 2 megs of memory. There are several versions of Prolog, such as Quintus, Turbo and Mprolog. Of these, Turbo runs on PC's, and requires only 0.5 megs of memory. Numeric languages . . . .**

The system proceeds by parsing the sentences. In the current implementation, the system can parse any sentence containing unknown nouns and adjectives, but verbs must be known to the the parser for a succesful parse to occur. If the parser fails then the system asks the user for a definition of those words which are unknown to the system. First of all, the system asks the user if the word in question is a verb (the user can check that by consulting a dictionary). If the word is a verb the system obtains the syntactical usage of the verb by activating an interface which allows users without a formal knowledge of English grammar to add new verbs to the parser. The system can make sense of many sentences containing words unknown to it. We postpone the discussion of how this is done until the next section. We briefly explain some of the formation rules used to construct the hierarchy in Figure 6. The first sentence does not create any structure. From the second sentence, the system creates the concepts *symbolic language* and *numeric language*, according to the techniques which we have explained for the formation of concepts denoted by restrictive clauses and noun groups. The *characteristic-features* slot of the concept *symbolic language* contains the conceptual relation "manipulate symbols." Note that from this moment on, the meaning of *symbolic language* is a

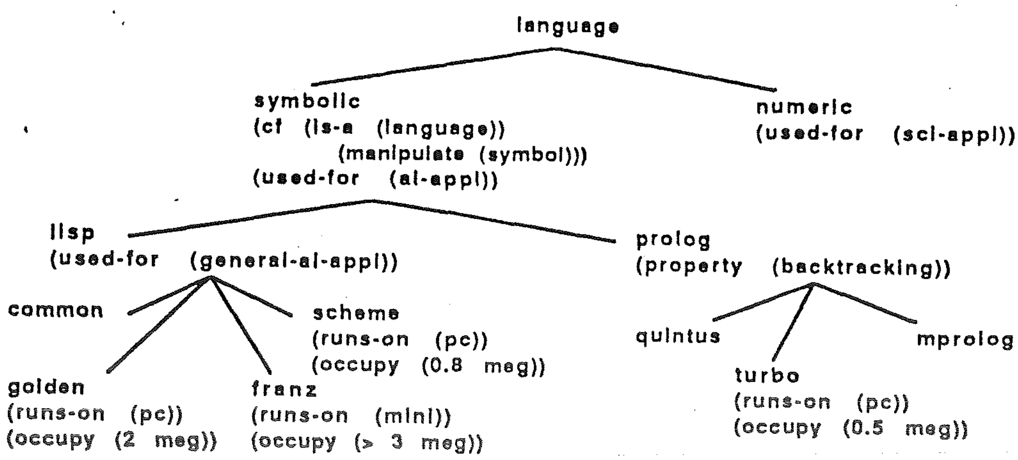


Figure 6: Hierarchy for the Text on Programming Languages

programming language which manipulates symbols. The formation of "symbolic languages are used for AI applications," is integrated by creating the descriptive relation "used-for (ai-application)" under the concept *symbolic language*. The next sentence creates two subconcepts of *symbolic language:* Lisp and Prolog. The next sentence differentiates the two languages. The formation rule which builds the LTM representation of "Prolog is used when backtracking is needed" constructs the following structure:

Prolog
  (property(backtracking))

This a general rule which forms all logical forms which fall under the schema:

<somebody> <use> <obj1> when/if/in case that <somebody> <need> <obj2>

into the structure:

obj1(property(obj2))

The rule is using the inference that if something, say *x*, is used when something else, say y, is needed then *y* is a property of *x*. This is necessary if requests such as "I need a symbolic language with backtracking" are to be answered by the consultation system.

5. Defining Unknown Words

One of the critical issues in the design of the system is how many concepts in the user's input need to be previously known by it in order for the system to understand the text and solve problems. If the knowledge acquisition system is going to be domain independent, then the system can not have *a priori* knowledge about concepts which are domain specific. For instance, the concept of "backtracking" in the sample text can not be part of the *a priori* concepts. However, if the knowledge acquisition system is going to deal with a concrete domain, then one can provide it with a rich set of domain dependent concepts. Since our goal is to build a domain independent knowledge acquisition system, the problem of unknown concepts becomes critical. The sample text can be processed by the present implementation without knowing anything about the following concepts: "programming languages," "symbols," "AI," "numeric languages," "backtracking," "Lisp languages," "meg" and "Lisp," Prolog," etc. If one asks the system "what is backtracking," this will reply by saying "I do not know what it is, but I know that it is a property of Prolog." The system's reply to a question is "I do not know" if the concept in the question is not linked to the concepts in the *a priori* hierarchy. For classification problem solving, the system does not need to have a deep understanding of what backtracking is. In any case, the decision of determining if a concept should be fully known to the system rests on the user. We will describe below some ways in which the user can achieve this.

Let us go back to the acquisition of the other concepts in the text. The mechanism for acquiring the concept "meg" differs from the acquisition of the other concepts in the paragraphs. The sentence in which the concept "meg" appears for the first time is "Franz ... takes more than 3 megs of memory." Let us assume that the system knows that memory is something where information is stored. The problem here is that when the system is going to form the final knowledge representation from the logical form of the sentence, it can not determine the

meaning of "takes" because it does not know the meaning of "meg." One of the meanings of "takes" is to occupy space or storage. A rule attached to "take" says: "if the object of "take" is unknown, but is numerically quantified then assume that the meaning of "take" is to occupy space." This rule will infer that meg is a unit of measurement and will ask the user to confirm its inference.

If a user deems that the system should have a concept classified in the hierarchy, or the system fails to build the representation of a sentence because some key concepts are missing, then the user can define these concepts by using the methods explained below.

Synonyms: This is the most linguistic and superficial of the methods. This method does not place concepts in the hierarchy. It is also the easiest to use, although, of course, it has limitations. For instance, the word "dialect of" in the middle of the paragraph may be unknown to the system. An easier way to define this word is to say that it is a synonym of "kind of," which the system clearly knows because it appears in the first sentence of the paragraph. Note that the words need not to be synonyms in all contexts, but just in the one the user is dealing with.

Classification: The best way to define a word is by providing the upper-concept of the concept denoted by the word to be defined. For instance, the word *AI* can be defined by saying "AI is a computer science." An inappropriate definition will be to say something like "AI tries to make computers intelligent," since this does not place *AI* in the hierarchy of concepts which forms the basis of SNOWY's understanding process. Likewise, the definition the system is expecting of *backtracking* is not a long and intricate one, which probably will be irrelevant to the expert system being built, but just something like "backtracking is a programming technique" which places *backtracking* in the hierarchy of concepts. We are extending the *a priori* hierarchy of categories to include a large set of concepts so that it will be easy to place new concepts in the hierarchy by classifying them.

Part-of Definitions: Some words may be defined by using a *part-of* relation. For instance, the word *cpu* or *computer memory* can be defined by saying that they are part of a computer.

Functional Definitions: We are studying short functional definitions of concepts from which the system can infer the *part-of* relation plus some other information. For instance, the concept *computer memory* above can be defined as: "memory is something to store information into a computer." From this, the system can infer that *computer memory* is a physical thing, part of a computer and is used to store information. The reason why this method of definition has not been incorporated in the system, yet, is because it opens the door for a user to give a definition too complex to be understood.

6. Problem-Solving

After SNOWY has read the paragraph of section 4, a user interested in a consultation about programming languages may, at this point, run a consultation test. For example, the user may say "I am interested in a language that runs on PC's and is good for ai applications." Given this inquiry, the task of the system is to find the languages that have the features "runs on PC's" and "used-for ai applications." The class of languages that have these features is represented as:

x1 (cf (is-a (language)) (runs-on (pc)) (used-for (ai-appl)))

If this concept were present in LTM, the question would be answered by simply retrieving its children. Since this is not the case, one way to find the answer to the question is to *classify* the concept *x1* ("languages that run on PC's and are good for ai") within the current hierarchy of concepts. This task is done by a *classifier algorithm,* which determines where in the current hierarchy a new concept, such as *x1*, would be located if it existed in LTM, that is, the *classifier* determines which of the concepts currently in LTM are the parents of *x1* and which ones are its children. We briefly illustrate the steps taken by the *classifier*, using the concept *x1* as an example. (See (Gomez and Segami, 1989a) for a detailed description of the algorithms in this section and a comparison to other classifier algorithms). Given the representation structure of *x1*, the concept within its *is-a* slot determines what part of LTM needs to be considered by the *classifier*. In this case, only the concept *language* or its children can be the parents or the children of *x1*. The *classifier* performs a traversal of selected nodes under the concept *language*. The concept in each visited node is compared with the concept *x1*. Depending on whether one is a subclass of the other, or there is no hierarchical relationship between them, the *classifier* decides which node to visit next. The actual comparison of concepts is performed by the *compare-classes* algorithm, which is based on the following idea. A concept *x2* is a subclass of a concept *x1*, if each characteristic-feature of *x1* is true of each entity in the class *x2*. In our example, concept *x1* is first compared with the concept *symbolic languages*. No hierarchical relationship is found between these two concepts. However, one of the *characteristic-features* of *x1*, "used-for ai applications," is found to be true of all the descendants of the concept *symbolic languages,* so that, when the nodes under *symbolic languages* are examined, only one more feature needs to be verified: "runs on PC's." After doing this, it is found that *scheme, golden,* and *Turbo* run on PC's. Then, the *classifier* returns the list "((language) (scheme golden turbo))" to indicate that the concept "languages that run on PC's and are good for ai" is a child of the concept *language* and that it has the children *scheme, golden,* and *Turbo*. The system, then, would reply to the user as follows: "the following languages run on PC's and are good for ai: scheme, golden, and turbo." If the user decides to narrow down his/her choices, he/she may either specify additional features, or formulate questions. For example, the following interaction may now take place:

user:

the language must run in a computer with less than 1 meg of memory

system:

scheme and turbo require less than 1 meg of memory

user:

what are the differences between scheme and turbo?

system:

scheme is a lisp dialect used for general ai applications, while turbo is a prolog dialect used for backtracking

The first sentence typed by the user simply adds one more requirement to the desired programming language. Since we have an initial set of solutions, the new requirement needs to be verified for the members of this set only. This reduces the possible solutions to two. The second sentence typed by the user is a question, which requires that a comparison be made

between the features of *Scheme* and *Turbo*. This is accomplished by traversing the hierarchy upwards, from the nodes corresponding to *Scheme* and *Turbo*. The traversal stops when the two paths cross each other.

The *classifier* is the component that allows SNOWY to maintain a correct organization of its memory. Each new statement entered by the expert may cause the insertion of a new concept in the hierarchy, or it may trigger a reorganization of the concepts in LTM. Let us suppose, for example, that the expert continues entering information to the system, by typing the following:

Some languages are procedural. Procedural symbolic languages are widely available.

The sentence "some languages are procedural" is integrated as explained in section 3.4. SNOWY's long-term memory configuration for the two sentences is depicted in Figure 7.

If the expert now types the statement: "Lisp is a procedural language," the concepts in LTM will be reorganized as shown in Figure 8.

Thus, in an inquiry of the form "a language that runs on PC's, is good for ai applications, and is widely available," only the Lisp dialects will be considered to be possible solutions.

As one can see the problem-solving method which we have illustrated uses only categorical knowledge (Szolovits and Pauker, 1978) and not probabilistic knowledge. It is our belief that probabilistic reasoning plays a minor role in most problem-solving situations if memory organization and indexing form part of the problem-solving method. In any case, certain aspects of probabilistic reasoning need to be included in our system. Experts use sentences

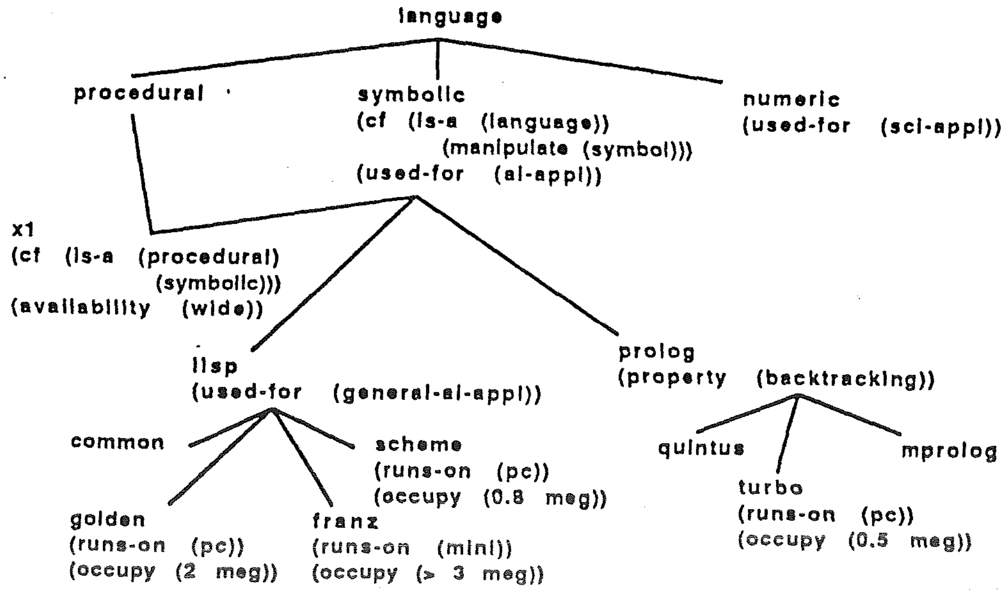


Figure 7: Organization of LTM after processing the passage "Some languages are procedural. Procedural symbolic languages are widely available."

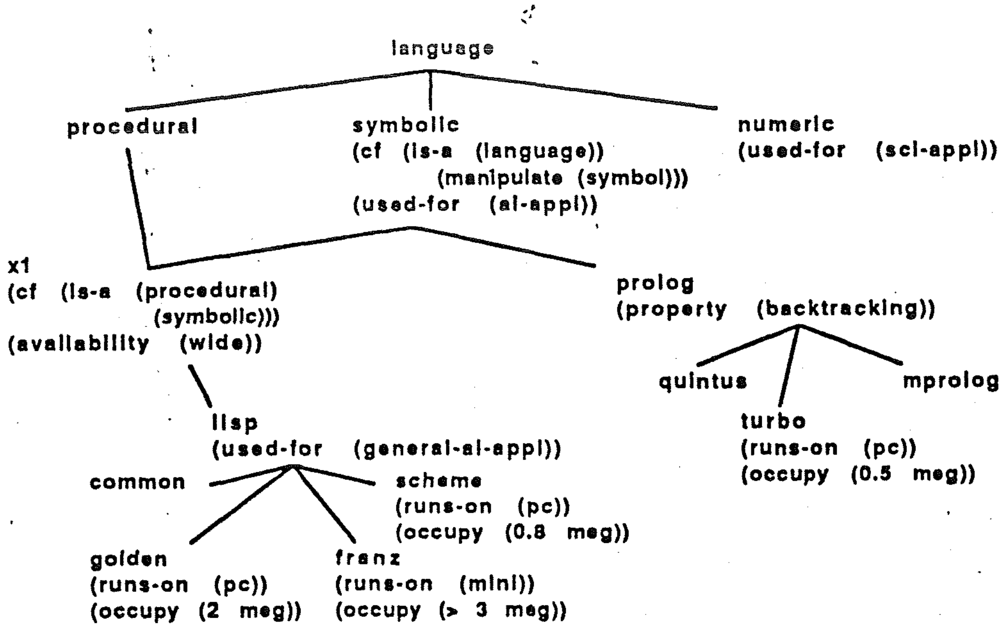


Figure 8: Reorganization of the concepts in LTM after processing the sentence "Lisp is a procedural language."

like this "Lisp is easier to learn than Pascal which in turn is easier to learn than Cobol." In cases like this, we need to ask the expert to rate Lisp, Pascal and Cobol with respect to the predicate *learn*. There are many ways in which this rating can take place and AQUINAS offers several ways which we plan to adopt in our system. Let us consider for the sake of the discussion that the expert produces the following rating: (Lisp (learn .9)) (Pascal (learn (.7)) (Cobol(learn (.4))). Then, the phrase "a symbolic language that is easy to learn" in the user's request "I am interested in a symbolic language which runs in a PC and is easy to learn," will be translated into a numeric value which may be greater than 5 according to the rating provided by the user for that predicate.

7. Discussion

In this paper, we have shown how a consultation system can be built from a natural language description by a human expert. It is our belief that the ideas and techniques described here can be used for the construction of expert systems for tasks which can be modeled using classification problem-solving. These tasks do not have to deal with consultation systems. They may be about diagnosis, design, etc. The relevance of the ideas presented in this paper go beyond the knowledge-acquisition area, since we have introduced a problem-solver which has striking differences from existing rule-based systems. In our system, organization and reorganization of memory, comprehension and problem solving are integrated processes. As a result, two important aspects of expert system technology become almost trivial: the updating of the knowledge base and interacting with the expert system. However, our approach is not without limitations. Let us first discuss the theoretical limitations. Although we believe that a

large class of problems and domains can be handled using our system, there are many domains which are beyond the reach of our method. The reason for that is that there are many aspects of the real world which can not be represented using the classification hierarchies we have presented in this paper. For instance, the functioning of complex systems such as the circulatory system, or the respiratory systems can not be adequately grasped using our present representation techniques. We are working in knowledge representation structures which will represent these systems from a point of view which will be adequate for comprehension and problem-solving (Gomez and Segami, 1989c).

There are many limitations to our present implementation. First of all, no expert has used yet our program to build a consulation system. All the consultation examples have been entered by graduate students or by the implementers of the system. Our system is a prototype which we can "demo" at any time, but that has not been used by any real user. This is clearly in contrast with AQUINAS which, according to the authors, has been used to build a variety of consultation systems. We are going to start to experiment with some real users to assess the real world capabilities of our system. Since SNOWY is able to acquire new concepts and new words, it can augment its lexicon by interacting with a user. The parser (Gomez, 1988) being used by SNOWY is a modular algorithm based on the notion of *syntactical usage of a word*, which allows a naive user without knowledge of English grammar to increase its syntactical coverage. We are writing an interface to our parser which will allow a user to add new verbs not presently in the parser. Yet, we are aware that there is a qualitative jump from testing a prototype by people who have certain familiarity with it to a real world system used by naive users.

We have presented a system and distinguished it from existing knowledge acquisition systems. Yet, our approach can be integrated with present knowledge acquisition tools and viceversa, techniques in other systems can be incorporated in our system. We think that, in future developments of our system for real users, the elicitation techniques developed for ETS and AQUINAS can be very useful. In particular, we are thinking that in those cases in which the English of certain users becomes too hard to handle, SNOWY may fall back on elicitation techniques. Likewise, domain dependent knowledge acquisition tools can benefit from a system which is able to build classification hierarchies from natural language text.

References

Boose, J. (1984). Personal construct theory and the transfer of human expertise. *Proceedings of the National Conference on Artificial Intelligence*, Austin, Texas.

Boose, J. and Bradshaw, J. (1987). Expertise transfer and complex problems: using AQUINAS as a knowledge acquisition workbench for expert systems. *International Journal of Man-Machine Studies*, 26(1), 3-28.

Charniak, E. and McDermott, D. (1984). *Introduction to Artificial Intelligence.* Addison-WesleyR, Reading, MA.

Clancey, W. J. (1985). Heuristic Classification, *Artificial Intelligence*, 27, 289-350.

Eshelman, L., Ehret, D., McDermott, J. and Tan, M. (1987). MOLE: a tenacious knowledge acquisition tool. *International Journal of Man-Machine Studies*, 26(1), 41-54.

Gomez, F. (1982). Towards a Theory of Comprehension of Declarative Texts, in *Proceedings of the 20th Meeting of the Association of Computational Linguistics*, Toronto, Canada, 36-43.

Gomez, F. (1985). A Model of Comprehension of Elementary Scientific Texts, in *Proceedings of the Workshop on Theoretical Approaches to Natural Language Understanding*, Halifax, Nova Scotia, 70-81.

Gomez, F. (1986). Knowledge Representation for Understanding Expository Texts, Technical Report CS-TR-86, Department of Computer Science, University of Central Florida, Orlando, Florida.

Gomez, F. (1988). WUP: A Parser Based on Word Usage, in *Proceedings of the 1988 IEEE Conference on Computers and Communication*, Scottsdale, Arizona, 445-449.

Gomez, F. and Chandrasekaran, B. (1984). Knowledge Organization and Distribution for Medical Diagnosis, in W. Clancey and E. Shortliffe (Eds.), *Readings in Medical Artificial Intelligence*. Addison-Wesley, Reading, MA.

Gomez, F. and Segami, C. (1989a). The Recognition and Classification of Concepts in Understanding Scientific Texts. *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 1, 51-77.

Gomez, F. and Segami, C. (1989b). Classification-Based Inferences in Retrieving Information from a Database of Scientific Facts. In *Proceedings of the Fifth IEEE Conference on Artificial Intelligence Applications*, Miami, Florida.

Gomez, F. and Segami, C. (1989c). Representing Systems as Event Hierarchies. Technical Report CS-TR-89-02, Department of Computer Science, University of Central Florida, Orlando, Florida.

Haas, N. and Hendrix, G. (1980). An Approach to Acquiring and Applying Knowledge. *Proceedings of the National Conference on Artificial Intelligence*, Stanford, CA, 235-239.

Kahn, G., Nowlan, S. and McDermott, J. (1985). MORE: an intelligent knowledge acquisition tool. *Proceedings of Ninth International Conference on Artificial Intelligence*, Los Angeles, California.

Marcus, M. (1987). Taking backtracking with a grain of SALT. *International Journal of Man-Machine Studies*, 26(4), 383-398.

Szolovits, P. and Pauker, S. (1978) Categorical and Probabilistic Reasoning in Medical Diagnosis. *Artificial Intelligence*, Vol. 11.

Appendix C

From: Proceedings of the First IEEE Computer Society International Workshop on *Tools for Artificial Intelligence, Fairfax, Virginia, 1989.*

Finding and Learning Explanatory Connections from Scientific Texts

Fernando Gomez and Carlos Segami
Department of Computer Science
University of Central Florida
Orlando, Florida 32816

Abstract

A theory for detecting and learning the explanatory connections between sentences in scientific texts is presented. A program embodying the theory is also described. The knowledge in the program is organized around the notions of analytic and empirical knowledge. Analytic knowledge encompasses very general rules which are valid across any domain, while empirical knowledge includes rules whose validity is domain dependent. Examples of these rules and their representation are given.

1. Introduction

In the last few years, we have proposed a model of comprehension of scientific or expository texts [3], and we have embodied the model into a computer program called SNOWY [6,7]. When SNOWY starts reading a scientific paragraph its knowledge about the world consists only of a bare hierarchy of concepts. As SNOWY reads, new concepts and relations are learned by integrating them in the right place in the hierarchy. For instance, after reading the passage below with no knowledge about birds, SNOWY acquires the concepts *bird, egreet, heron, birds which live in swamps*, and *birds which live in forests.*

> Birds are animals. Some birds, such as the egreet and heron, live in swamps. Others live in forests.

There are several components in SNOWY, as illustrated in Figure 1. First, the sentences are parsed into a case structure using a parser based on the the idea of syntactical usage of a word [4]. The output of the parser is passed to the Formation Phase, which consists of two components: interpretation and formation. During interpretation the logical form for the sentence is constructed resolving prepositional attachment, anaphoric reference, etc. During formation properly said, the final knowledge representation structures are built from the logical form of the sentence. These knowledge representation structures are passed to the Recognition Phase, which checks if the concepts which have been formed exist in long-term memory (LTM). Those concepts which the Recognition Phase fails to recognize are passed to the Integration Phase which integrates them in LTM upon activating a Classifier Algorithm that indicates where and how they must be integrated in LTM. Finally, the Explanatory Phase tries to understand the explanatory links connecting sentences (e.g. Antibiotics work against infections. They kill the germs which cause them.)

Since SNOWY is reading novel texts and assimilating new concepts and relations, the system embodies a knowledge-lenient model of comprehension, that is, one which does not involve schemas or frames to be instantiated as sentences are read. This differentiates SNOWY from knowledge-intensive models of comprehension based on scripts, plans or MOPs [2,12,13]. An essential aspect of comprehending in the face of limited knowledge is that of learning, since successful comprehension involves the learning of new concepts and relations. There are several aspects of learning involved in SNOWY. The first one is that of acquiring classification hierarchies from texts [5]. For instance, from the sentences below

> Not all animals which live in the sea are fish.
> Some are mammals.

Snowy automatically constructs two subclasses of "animals," namely, "animals which live in the sea" and "fish," and two subclasses of "animals which live in the sea," namely, "sea fish" and "sea mammals." There are, of course, many other examples of sentences in which classification hierarchies must be learned if understanding is to occur.

A second aspect of learning involved in SNOWY is that of failure-driven learning. This occurs when SNOWY fails to recognize a relation or concept. For instance, if the system reads "antibiotics kill germs," knowing "antibiotics," but not "germs," the system will infer (from the concept "kill") that "germs" is an animate and will learn the relation underlying the sentence by integrating its conceptual representation in LTM.

However, in this paper we will present a third aspect of learning which is essential in understanding scientific prose, namely *explanation-driven* learning. In the next section, we introduce the concept. In section 3, we discuss the notions of analytical and empirical rules. Section 4 deals with the representation of concepts in SNOWY, and section 5 describes how analytical and empirical rules are activated. Finally, section 6 contains the conclusions.

2. Explanation-Driven Learning

Describing and explaining are two essential characteristic of scientific prose. Classification and the introduction of new concepts

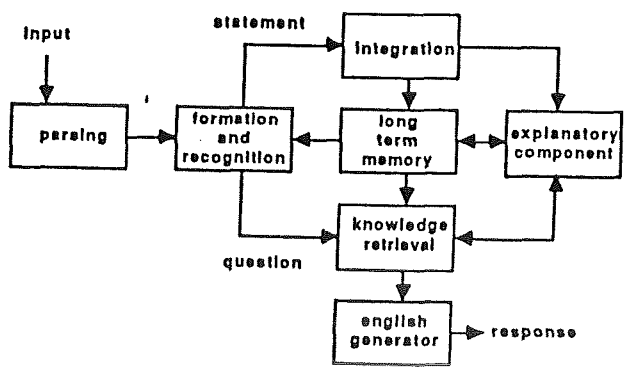


Figure 1. The Components of SNOWY

(*) This work was partially supported by NASA, Kennedy Space Center, under grant No. NAG-10-0058.

and relations are aspects of describing. Explaining is also an important aspect of scientific texts. This notion bears only a weak resemblance to the notion of explanation-based learning, which has recently become popular in research on learning [1,8]. Our notion of explanation was first presented in [3], and is based on the observation that there are two clearly distinct types of texts, which we have called explanatory and descriptive. An example of descriptive text is:

> An alveolar air sac looks like a bunch of balloons. The walls of each sac are very thin and moist. Alveoli are surrounded by a rich suply of capillaries.

In descriptive texts, concepts are described in terms of new or old concepts. In these texts, descriptive verbs and predicates predominate over action verbs. However, in explanatory texts sentences are explanations of relations introduced by previous sentences. Consider the following passages:

> (1) Antibiotics work against infections. They kill the germs which cause them.
>
> (2) Lions are very powerful. Few animals dare to bother them.
>
> (3) Plants are dying. It is 20 degrees F outside.

In (1), the second sentence is an explanation of how antibiotics work against infections. In (2), the second sentence explains a consequence of lions being powerful. In (3), the second sentence is an explanation of why plants are dying. The understanding of these sentences requires the learning of the explanatory links which connect them. Without the acquisition of these links, one would not be able to answer the questions: How do antibiotics work against infections? Why do few animals dare to bother lions? or Why are plants dying? We have identified the following links between sentences. See [3] for a detailed description of theses links.

1. Why do animate beings do things?
2. Why are animate beings in states?
3. What is the result or effect of states on those which suffer them?
4. Why are the objects of actions recipients of those actions?
5. What is the result or effect of physical actions on the objects of those actions?
6. What is the result or effect of actions on the instruments or recipients of those actions?
7. How are actions performed or how do they take place?
8. How much time do actions take?
9. How do animate beings behave or react to the actions performed by other animate beings towards them?
10. Why do certain things have certain properties?
11. What is the result or consequence of something having certain properties?
12. Why do certain things happen to be at certain places?
13. How small, large, etc. are things? The link consists of specifying fuzzy adjectives and adverbs. For example, "Plankton are tiny animals. Most of the plankton are one-thousandth of an inch."

3. Learning the Connections

There are two types of knowledge which allow the system to learn the links connecting sentences, which we have called analytical and empirical knowledge, analogous to the notions of analytical and empirical sentences. An analytical sentence is one whose truth conditions can be established by only having knowledge about the language. Classical examples of analytical sentences are "all bachelors are unmarried" and "all white horses are horses." The determination of the truth conditions of empirical sentences, on the other hand, requires knowledge about the world. For instance, the sentence "plants need light to live" requires knowledge about plants and light to establish its truth. We have used two types of rules to encapsulate analytical knowledge and empirical knowledge, which we call analytical and empirical rules, respectively. In our opinion, Quine [10] has argued convincingly that there is not a dichotomy between analytic statements and empirical statements, but that the difference is one of degree. Yet, we consider that this distinction, although not dichotomous, is a relevant one, because we want to find out how much of the learning of new scientific concepts is based on analytical rules, and how much is based on empirical rules. The reader may find helpful to view analytical rules as very general common sense rules. Analytical rules are anchored in each primitive verbal concept. Some examples of analytical rules are:

> (1) if ingest(x,y) then enter(y,x)
> (2) if ingest(x,y) and located-on(z,y) then ingest(x,z)

The negation of rule (1) will result in a flagrant contradiction, one in which the meaning of "ingest" will be changed. Regarding (2), it is hard to imagine a world in which (2) does not hold. Let us consider the relation between analyticity and learning. Every time that an analytical rule is applied in order to find the connection between two sentences, a new piece of knowledge is learned. Consider the passage:

> Antibiotics work against infections. They kill or slow down the germs which cause infections.

If after reading the first sentence we are asked "how do antibiotics work against infections?" we do not know the answer. Yet, after we read the second sentence, we can answer the question. This answer is based on an analytical rule which says:

> if destroy(x,y) and cause(y,z) then work-against(x,z)

(where x is instantiated to antibiotics, y to germs, and z to infections). Note that the antecedent or premise of the rule is the explanation of how antibiotics work against infections. Then, the analytical rules are not only making possible understanding but also learning. In the above example, a new empirical fact has been learned, namely, that antibiotics are effective against infections because they destroy the germs which cause them. The analytical rule which has been used is of a general nature and, consequently, can be applied in many other different situations. Consider the passage:

> During the San Francisco earthquake most houses burned. They were made of wood.

The connection can only be understood if we have an empirical rule which says that "if an object is made of wood, it burns." This notion of empirical rule is more manifest in the sentences:

> Plants are dying. It is freezing outside.

It is generally accepted that it is a law of nature that if plants are submitted to temperatures below zero, they are affected negatively. The empirical nexuses connecting sentences are also found in the societal world, as exemplified in following sentences:

> Lions are very powerful. Few animals dare to bother them.

The rule connecting these two sentences is not only true of lions, but also of politicians, countries, etc. In all these examples, the knowledge which connects the sentences can be considered to be domain specific common sense knowledge. Yet in other cases, very specialized knowledge may be needed to establish the connection. An entire scientific theory may mediate between the two sentences, as in the examples below:

> There are worms in the meat. They grew spontaneously.
> Cars are cheap. They are mass produced.

The first example invokes the theory of spontaneous generation to explain the appearance of worms in the meat. In the second example, a theory of mass production is needed for a thorough explanation of the link connecting the sentences.

Our present implementation includes a small set of empirical rules dealing with plants and animate beings in general. These rules are stored into the concepts in the hierarchy so that they can be inherited by children concepts.

4. Representation of Concepts

In this section we describe briefly the knowledge representation structures used in SNOWY. A detailed discussion can be found in [6]. Concepts are represented in LTM in frame-type structures, which are linked through *is-a* and *classes-of* slots to yield a hierarchical organization. Two kinds of representation structures may be part of LTM: *object-structures* and *action-structures*. An *object-structure* contains knowledge about physical or abstract objects and has slots called conceptual relations, or attributes. Two kinds of conceptual relations can be distinguished: those which describe objects and those which attribute actions to physical things. Thus, "centipedes are arthropods" and "all centipedes are slim and have many legs" are descriptive sentences, while "antibiotics kill germs" and "cells take in sugar" express actions performed by physical things. Descriptive attributes are represented following the frame notation. For instance, after reading the two sentences about centipedes, the *object-structure* built for *centipede*, will be the following:

```
(centipede
   (is-a (arthropod))
   (form (slim (q1 (all))))
   (has-body-part (leg (q1 (all)) (q2 (many)))))
```

The slot *q1* contains the quantifier of the concept described by the *object-structure*, and the slot *q2* contains the quantifier of the second argument of the relation. Conceptual relations denoting actions, on the other hand, are represented through *action-structures*. For example, the relation underlying the sentence "antibiotics kill germs" is represented by the *action-structure*:

```
(@a00001
   (args (antibiotics) (germs))
   (pr (cause-to-die))
   (q1 (?))
   (q2 (?)))
```

The slot *args* contains all the arguments of the relation. The first concept in this slot is always the actor of the action verb and the second concept is the object. The slot *pr* contains the relation (*pr* stands for *primitive-relation*). The slots *q1* and *q2* contain the quantifiers of the first and second arguments, respectively. The term *@a00001* is a symbol generated by the system, which stands for the name of the conceptual relation. *Action-structures* are linked to all the arguments of the relation. Thus, the complete representation of the sentence "antibiotics kill germs" consists of the following *object-structures* and *action-structure:*

```
(@a00001 (args (antibiotic) (germ)) (pr (cause-to-die))
         (q1 (?)) (q2 (?)))
(antibiotic (cause-to-die (germ ($more (@a00001)))))
(germ (cause-to-die:by (antibiotic ($more (@a00001)))))
```

Note how the *object-structures* of both, *antibiotic* and *germ*, point to the *action-structure* @a00001.

Let us now consider the sentences

> Birds with long legs live in swamps. They have a long bill to extract food from under water.

This text introduces the concept "bird with long legs" for which a name is not given. However, a unique name in memory needs to be created for this concept so that additional knowledge about it can be learned by storing this knowledge under the same node, and so that this node can be linked to the other nodes in the hierarchy. Restrictive relative clauses and complex noun groups are represented in LTM via *object-structures* with a dummy name (a gensym), and their representation is characterized by the presence of a *characteristic-features* slot whose purpose is to identify the concept by describing the features that are characteristic of it. For example, when building the representation of the sentence, "Insulin is a hormone produced by the pancreas," the concept *hormones produced by the pancreas*, a subconcept of the concept *hormones*, is created with the following representation:

```
(x1 (cf (is-a (hormone)) (make:by (pancreas (q2 (?))))))
```

Here, we use *cf* to denote *characteristic-features*. Within the *characteristic-features* slot there is always at least an *is-a* slot, which specifies an ancestor node of the concept being represented. In this example, the concept x1 has been created as a subconcept of *hormones*, and further characterized by the fact that they are made by the pancreas. The newly-created concept is identified by the content of the *characteristic-features* slot and not by the dummy name "x1." Thus, if the question "what do hormones produced by the pancreas cause?" is asked, the system recognizes that the concept *hormones produced by the pancreas* was previously created and accesses the *object-structure* x1. (The slot *q2* contains "?" because it is unknown if every pancreas makes hormones.) Similarly, the representation of *bird with long legs* proceeds by creating the concept *long leg* first, and then the concept *bird with long leg*, as illustrated below.

```
x1 (cf (is-a (leg)) (length (long)))
x2 (cf (is-a(bird)) (has-body-part (x1 (q2 (?)))))
```

5. Activating the Rules

We now illustrate how the analytical and empirical rules are activated by following through two examples. Suppose the system reads the sentence

> The human body recognizes the germs which enter it.

First, the sentence is parsed. The parser builds structures which identify the syntactical parts of the sentence. In our example, the parser output is the following:

```
g00008
(subj ((dfart the) (adj human) (noun body)) verb (recognizes)
      obj ((dfart the) (noun germs)) rela g00009)
g00009
(subj ((dfart the) (noun germs)) verb (enter) obj ((pron it)))
```

These structures become the input to the Formation Phase, which goes through the interpretation and formation components in order to build the representation structures of the concepts and relations underlying the given sentence. In our example, the Formation Phase returns the following *object-structures* and *action-structures*:

```
(@a00009 (args (DMY00008) (human-body))
   (pr (enter)) (inside (human-body))
   (q1 (all)) (q2 (?)) (medium (?)) (how (?)) (source (?)))
(human-body (enter:by (DMY00008 ($more (@a00009))))
   (become-aware (DMY00008 ($more (@a00010)))))
(DMY00008 (enter (human-body ($more (@a00009))))
   (cf (is-a (germ)) (enter (human-body (inside (human-body)))))
   (become-aware:by (human-body ($more (@a00010)))))
(@a00010 (args (human-body) (DMY00008)) (pr (become-aware))
   (q1 (?)) (q2 (?)) (how (?)))
```

Figure 2. Representation structures built by the Formation Phase after reading the sentence "the human body recognizes the germs which enter it"

After completing the phases of parsing, formation and recognition, the Integration Phase inserts the newly built *object-structures* and *action-structures* into LTM. The integration of *object-structures* involves its classification among the concepts in LTM and, in some cases, they may trigger a reclassification of the concepts currently present in LTM. See [6] for a detailed discussion of this. When an *action-structure* is integrated in LTM, the integration component examines the definition of the primitive in this structure to determine which of its case slots have not been filled by the sentence just read. The unfilled case slots are placed in a list, a list of explanatory links, which is passed to the explanatory component. The explana-

tory links generated for the sentence of our example are the following:

((medium enter (DMY00008 human-body) @a00009)
(source enter (DMY00008 human-body) @a00009)
(how enter (DMY00008 human-body) @a00009)
(how become-aware (human-body DMY00008) @a00010))

Here, three links were generated for the action "germs enter the human body," and one link for "the human body recognizes DMY00008." The symbol *DMY00008* corresponds to the concept "germs which enter the human body." The symbols *@a00009* and *@a00010* are the names of the *action-structure* for which the corresponding link was generated. Once the explanatory component is activated, its first task is to determine if the sentence just read answers any of the links generated by previous sentences. These links are kept in the list **links, which at this point is empty, since we have read our first sentence. Next, the explanatory component examines the list of links just passed by the integration component, in order to determine if it is possible to answer any of them by using the analytical and empirical rules and the current knowledge in LTM (This situation occurs, for example, if the sentences "Lions are very powerful. Few animals dare to bother them" are read. In this case, the link "why do few animals dare to bother lions?" which would be generated from the second sentence, is answered by the first sentence). After doing this, the links that were passed to the explanatory component which are left unanswered are stored in the list **links. In our example, all four links shown above will be stored in **links.

This completes the processing of the first sentence. Let us now suppose that the next sentence read by the system is the following:

These germs are found in the food humans eat

After inserting the new concepts and relations in LTM, the integration component generates the links shown below, and activates the explanatory component.

((how ingest (human DMY00016) @a00017)
(source ingest (human DMY00016) @a00017)
(dest ingest (human DMY00016) @a00017))

These links correspond to the action "humans eat DMY00016," where the symbol DMY00016 corresponds to the concept "food eaten by humans." The explanatory component now examines the representation structures built by the formation phase for the current sentence. In our example, the structures to be examined are the following:

```
(@a00017 (args (human) (DMY00016)) (pr (ingest))
        (q1 (?)) (q2 (all)) (how (?)))
(DMY00016 (ingest:by (human ($more (@a00017))))
        (cf (is-a (food)) (ingest:by (human)))
        (location-of (DMY00008)))
(human (ingest (DMY00016 ($more (@a00017)))))
(DMY00008 (location (DMY00016)))
```

Figure 3. Representation structures built by the Formation Phase after reading the sentence "These germs are found in the food humans eat"

For each relation introduced by the sentence, the analytical rules attached to its primitive are retrieved. In this case, the first relation examined is the action *@a00017*, so the rules attached to *ingest* are retrieved. One of these rules is the following:

(((location ($x $y))(ingest ($z $y))) (medium enter ($x $z)))

The rule may be read as follows: "if x is located in y, and z ingests y, then x enters z through y." Once the analytical rules are retrieved, each link in **links is matched against the *then* part of the rules. At this point, after reading the two sentences above, the list **links contains:

((medium enter (DMY00008 human-body) @a00009)
(source enter (DMY00008 human-body) @a00009)
(how enter (DMY00008 human-body) @a00009)
(how become-aware (human-body DMY00008) @a00010))

As we can see, the first link in **links matches the *then* part of the above rule, by instantiating variable $x to *DMY00008* and variable $z to *human-body*. Because there is a match, the explanatory component tries to verify the *if* part of the rule which, after instantiating the variables $x and $z, has the form:

((location (DMY00008 $y)) (ingest (human-body $y)))

To verify these two conditions, the explanatory component activates the knowledge retrieval component [7]. Each condition is passed to the knowledge retrieval component in the form of a question. Thus, the first condition becomes the question "where are germs which enter the human body found?" Since the knowledge the system has about germs is what has been specified in our two sentences, the answer to this question will be "germs which enter the human body are found in the food humans eat," thus instantiating the variable $y to DMY00016 (food humans eat). Then, the second condition in the *if* part becomes "(ingest (human-body DMY00016))," which is passed to the knowledge retrieval component as the question "does the human body ingest DMY00016?" Since the answer to this question is affirmative, the rule has fired successfully, providing an answer to the explanatory link "(medium enter (DMY00008 human-body) @a00009)." The explanatory component then proceeds to fill the *medium* case slot in the *action-structure* @a00009 with the concept DMY00016 (food humans eat), and to remove the link "(medium enter (DMY00008 human-body) @a00009)" from **links. The rest of the links are examined in a similar way.

Let us now briefly consider a second example. Suppose the system reads the sentences:

Plants are dying. It is 20 degrees F outside.

After reading the first sentence, the explanatory link "(why negative-state(plant))" is generated. The answer to this link is obtained after reading the second sentence, by activating the empirical rule:

if less-than(temperature 32F) then negative-state(animate-being)

The rule encodes the fact that if the temperature is below 32 degrees, animate beings enter a negative-state (a semantic feature associated with such states as *death, infection*, etc.). The rule is stored under the concept *animate-being*, and inherited by the concept *plant* through the *is-a* hierarchy.

Using the Rules to Answer Questions

Let us consider the sentences

Antibiotics work against infections. They kill the germs that cause infections.

When the first sentence is read by the system, an *action-structure*, say *@a1*, will be built to represent the underlying relation, and one of the explanatory links generated will be "(how work-against (antibiotic infection))." This explanatory link will be answered by the second sentence because one of the analytical rules attached to the primitive *cause-to-die* is

(1) if cause-to-die(x y) and cause(y z) then (how (work-against(x z)))

which will fire successfully. Thus, the *how* case slot of the *action-structure @a1* will be filled with the relation "cause-to-die (antibiotic x1)," where x1 is the concept "germs which cause infections." If the question "how do antibiotics work against infections" is now asked, a simple access to the structure *@a1* in LTM will provide the answer. Suppose, on the other hand, that the system first reads the sentence:

Antibiotics kill the germs that cause infections

As before, once the knowledge in this sentence is integrated in LTM, the explanatory component retrieves the analytical rules attached to the primitives in the sentence, and fires them in an attempt to answer previously generated links. One of the rules retrieved is rule (1), above. However, contrary to our previous

example, this time after reading the sentence "antibiotics kill the germs that cause infections," the explanatory component will not add any new knowledge to LTM. The reason is that, since no sentences have been read before, there are no explanatory links to be answered. Therefore, questions like "how do antibiotics work against infections?" or "do antibiotics work against infections?" cannot be answered by the system. An obvious solution to this problem would be to fire all empirical and analytical rules in a forward fashion each time that a new sentence is integrated, and to add to LTM each piece of knowledge obtained. There are two problems with this solution. First, it takes a lot of processing time. A more serious problem, however, is that many irrelevant pieces of knowledge would be added to LTM. The solution we have adopted for this problem is the following. If the knowledge retrieval component does not find in memory the answer to a given question, it activates the explanatory component to try to infer the answer from memory. The explanatory component selects the analytical rules attached to the primitive of the verb in the question, and also the empirical rules, if any, attached to the main concepts in the question. This is what will happen if the question "do antibiotics work against infections?" is asked. The Explanatory Component gets the analytical rules attached to *work-against* and selects those whose *then* part have "work-against" in them (because the question has the primitive work-against). Then it tries to verify the truth of the *if* part of those rules. In our example, since rule (1) above is attached to *work-against*, as well as to *cause-to-die*, the Explanatory Component will instantiate this rule as:

> if cause-to-die(antibiotics y) and cause(y infections) then work-against(antibiotics infections)

Then, in order to verify the truth of the *if* part of the rule, the Explanatory Component, in turn, activates the Knowledge Retrieval component. The Explanatory Component formulates the question "what do antibiotics cause-to-die?" The Knowledge Retrieval component will return a list containing the things killed by antibiotics. Then, the Explanatory Component instantiates the variable *y* to each of the elements in the list and, again, asks the Knowledge Retrieval component if any of those things cause infections. If the antecedent of the rule is found to be true, the answer to the question "do antibiotics work against infections" will be *yes*. The method we have explained allows us to obtain an analytical implication only when it is required to answer a question.

6. Related Research and Conclusions

There are striking differences between the type of learning described in this paper and explanation-based learning (EBL). In [8], the following four kinds of conditions need to be specified for EBL to occur.

(1) *Necessary* and *sufficient* conditions expressing the concept to be learned.

(2) A training example describing a positive example of the concept to be learned.

(3) A *domain theory* consisting of a set of facts and rules which explain how training examples are instances of the concept to be learned.

(4) An *operationality criterion* which specifies a set of predicates in which the concept to be learned needs to be reexpressed.

First of all, the algorithm explained in this paper does not learn concepts, but relations or connections between concepts. Secondly, and more importantly, the relations to be learned are not given to our system through the specification of necessary and sufficient conditions. The requirement that concepts to be learned be specified by defining them using necessary and sufficient conditions, constitutes, in our opinion, one of the most serious limitations in the current learning research. Our approach explores the learning of the connections which link the concepts in a web with no specific layers (see [11]).

In contrast to EBL, there is no training example in our method. Relations between concepts are learned as they are given in the text. Of course, the performance of our algorithm degrades if many sentences mediate between the introduction of a conceptualization and its explanation, something that occurs in poorly written texts in which explanations are zigzagging, causing it to be difficult, even for good readers, to understand the text being read. These are issues we need to explore further and provide some quantifications for the concepts of "algorithm degradation" and "many mediating sentences."

There is a point of coincidence between requirement (3) in EBL, the domain theory, and our analytical and empirical rules. The main differences lie in the fact that, in EBL, the domain theory consists not only of rules, but also of facts. However, our domain theory consists only of rules. An important distinction is introduced in our domain theory by classifying the rules into analytical and empirical rules.

Finally, there is not a subset of predicates which constitutes the operationality criterion in our algorithm. A connection between two concepts is learned if predicates (not just a selected set of them) can be found in LTM which satisfy a rule or rules in the domain theory.

We have shown how a program can find and learn the explanatory connections existing between sentences. Our present implementation contains about forty analytical rules and ten empirical rules. Analytical rules are general in nature and are applicable to diverse domains. In fact, they are an essential component to the understanding of any domain. Empirical rules are domain dependent and do not play as essential a role as analytical rules.

Whether the distinction between analytical and empirical statements has any cognitive relevance has been the object of much debate in the philosophical and psychological literature. The study of Piaget and his coworkers [9] indicates that although human subjects do not "feel" a strong demarcation between these two types of sentences, yet they do sense that a relevant distinction is present in most cases. The preliminary feedback we have obtained from our program indicates that analytical and empirical knowledge are necessary elements in finding and learning the explanatory links which connect sentences in scientific discourse. From an engineering point of view, this distinction provides a relevant criterion for organizing the knowledge of a program in two distinct, although not dichotomous, categories. In our ongoing research, we are investigating the role that this distinction of analytical and empirical knowledge plays in knowledge-based problem solving.

References

[1] DeJong, G. F. and Mooney, R. J. "Explanation-Based Learning: An Alternative View," *Machine Learning, 1,2, 1986.*

[2] Dyer, M. G. *In-Depth Understanding.* MIT Press, Cambridge, MA, 1983.

[3] Gomez, F. A Model of Comprehension of Elementary Scientific Texts, In *Proceedings of the Workshop on Theoretical Approaches to Natural Language Understanding,* Halifax, Nova Scotia, 70-81, 1985.

[4] Gomez, F. WUP: A Parser Based on Word Usage, In *Proceedings of the 1988 IEEE Conference on Computers and Communication,* Scottsdale, Arizona, 445-449, 1988.

[5] Gomez, F. Learning by Being Told: The Acquisition of Classification Hierarchies. Technical Report CS-TR-88-06, Department of Computer Science, University of Central Florida, Orlando, Florida, 1988.

[6] Gomez, F. and Segami, C. The Recognition and Classification of Concepts in Understanding Scientific Texts. *Journal of Experimental and Theoretical Artificial Intelligence,* Vol. 1, 51-77, 1989.

[7] Gomez, F. and Segami, C. Classification-Based Inferences in Retrieving Information from a Database of Scientific Facts. In *Proceedings of the Fifth IEEE Conference on Artificial Intelligence Applications,* Miami, Florida, 1989.

[8] Mitchell, K. M., Keller, R. and Kedar-Cabelli, S. "Explanation-Based Generalization: A Unifying View," *Machine Learning 1,1. 1986.*

[9] Piaget, J. (ed.) Les liaisons analytiques and synthetiques dans le comportement du sujet. *Etudes d'epistemologie genetique.* Paris, 1959.

[10] Quine, V. W. *From a Logical Point of View,* Harvard University Press, 1953 (Revised 1963).

[11] Quine, V. W. *Word and Object,* MIT Press, 1960.

[12] Schank, R. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, 1982.

[13] Wilensky, R. *Planning and Understanding.* Reading: Addison-Wesley, 1983.

APPENDIX

SHORT SAMPLE SESSION WITH SNOWY

In this sample session, lines preceded by ">>>" correspond to the user's input to SNOWY. Other lines, except for the remarks, correspond to SNOWY's response.

>>> the human body fights infections.
>>> the human body kills the germs that cause infections.
>>> how does the human body fight infections?

human-body work-against infection
how <human-body cause-to-die <germ which cause infection>>

REMARKS: After the integration of the second statement above, the Explanatory Component answers the link "how does the human body fight infections?," generated by the first statement, by firing the analytical rule "if x kills y and y causes z then this explains how x fights z." The explanation found is inserted in the appropriate *action-structure* in LTM, and the system is then able to answer the question "how does the human body fight infections?," as shown above.

>>> insulin controls the level of sugar in the blood.
>>> does insulin control sugar in the blood?

::: yes
insulin control <sugar which location blood>
because
insulin control <level pertaining-to <sugar which location blood>>

REMARKS: Here, the Knowledge Retrieval Component is unable to answer the question by accessing the current knowledge in LTM. Therefore, it activates the Explanatory Component, which fires the analytical rules attached to the primitive *control*. One of these rules is "if x controls y and y pertains to z, then x controls z." This rule fires successfully, and the system is able to answer the question, as shown above.

>>> the human body produces macrophages which kill bacteria.
>>> does the human body kill bacteria?

::: yes
human-body cause-to-die bacteria
because
human-body make <macrophage which cause-to-die bacteria>

REMARKS: This situation is similar to the previous one. Not being able to find an answer in LTM, the Knowledge Retrieval Component activates the Explanatory Component, which fires the rule "if x produces y and y kills z, then x kills z."

>>> bacteria cause infections to humans.
>>> infections kill humans.
>>> do bacteria kill humans?

::: yes
bacteria cause-to-die human
because
infection cause-to-die human and
bacteria cause infection *recipient* human

REMARKS: The analytic rule that allowed this question to be answered is "if x causes y to z and y kills z, then x kills z."

>>> germs which enter the human body cause diseases.
>>> these germs are found in the food humans eat.
>>> the human body kills these germs by producing antibodies.

REMARKS: The second statement provides the answer for the link "how do germs enter the human body?" generated by the first statement, and this new knowledge is inserted in LTM after integrating the second statement. Then, the system is able to answer the question below by accessing the LTM structures.

>>> how do germs enter the human body?

germ enter human-body *medium* <food ingest:by human>

>>> does the human body fight diseases?

::: yes
human-body work-against disease
because
human-body cause-to-die <germ which enter human-body> and
<germ which enter human-body> cause disease

REMARK: The answer to this question is found by activating the analytical rule "if x destroys y and y causes z, then x work-against z."

>>> do antibodies kill germs?

::: yes
human-body cause-to-die germ *how* <human-body make antibody>

REMARKS: In this case, the analytical rule applied is "if x kills y by producing z, then z kills y."

Appendix D

Knowledge Acquisition (1990) **2**, 107–128

Knowledge acquisition from natural language for expert systems based on classification problem-solving methods

FERNANDO GOMEZ AND CARLOS SEGAMI

Department of Computer Science, University of Central Florida, Orlando, FL32816, USA

(*Received 1 September 1989 and accepted in revised form 27 February 1990*)

It is shown how certain kinds of domain independent expert systems based on classification problem-solving methods can be constructed directly from natural language descriptions by a human expert. The expert knowledge is not translated into production rules. Rather, it is mapped into conceptual structures which are integrated into long-term memory (LTM). The resulting system is one in which problem-solving, retrieval and memory organization are integrated processes. In other words, the same algorithm and knowledge representation structures are shared by these processes. As a result of this, the system can answer questions, solve problems or reorganize LTM.

1. Introduction

Can a simple expert system consisting of a hundred rules be designed from a natural language description by a human expert? Or can an expert system be updated from a natural language description of some of its components? In this paper, we present an integrated and domain independent system which builds classification-hierarchies from texts, retrieves information from them and solves problems.

We have been investigating the relationship between comprehension and problem-solving. Our investigation has been guided by the idea that problem-solving and comprehension are not two separate processes each one with its own knowledge representation structures, but that they spring from a common source and share the same knowledge representation structures. It is beyond the scope of this paper to discuss these issues in detail, rather we centre our discussion in providing evidence on how certain kinds of expert systems can be built from a natural language description by a human expert. We will show that the problem solving method used by the expert system to solve problems is the same as that used by the comprehender to keep memory organized and answer questions requiring chains of inferences.

We can roughly characterize the domain of the expert systems to which we have applied our method as consultation tasks. For instance, the tasks of building an expert system to select a programming language, or select a physician, etc. Some of these tasks are described in (Boose & Bradshaw, 1987), We can further characterize the scope of our system by the problem-solving method used to find solutions to a problem. The problem-solving method is that of classification

This research is supported by NASA-KSC Contract NAG-10-0058.

1042-8143/90/020107 + 22$03.00/0

problem-solving (Gomez & Chandrasekaran, 1984; Clancey, 1985). Classification-problem solving consists of conceptualizing the domain of the problem-solver into a hierarchy of concepts and designing an algorithm which solves a problem by placing it in the hierarchy of concepts. In Gomez and Chandrasekaran (1984) it is stated:

> Given these principles of organization of medical knowledge, the solution of a medical case becomes a problem of taxonomic classification. It is similar to the problem of placing, say, a specimen of maple in a hierarchy of botanical concepts. It consists of identifying its superordinate and subordinate concepts.

The method to recognize problem solutions by placing them in a hierarchy of concepts was that of using clusters of production rules organized under the concepts in the hierarchy, called *specialists* (Gomez & Chandrasekaran, 1984). In this paper, (a) we will show how to automatically build the classification hierarchies from natural language and (b) we will describe an algorithm which does not use the production rules to place the problem solutions in the hierarchy.

Among the knowledge acquisition systems which have been developed, ETS (Boose, 1984) and AQUINAS (Boose & Bradshaw, 1987) are the closer to our approach in the sense that these systems are domain independent and are based on building classification hierarchies for the solution of the problems. However, the differences are remarkable since we are not using elicitation techniques to build the classification hierarchies, but we are building them from natural language descriptions. Also, we are not translating the expert's knowledge into production rules, but we are using a general algorithm to produce solutions from the constructed hierarchies. Nano-KLAUS (Haas & Hendrix, 1980) was an earlier natural language acquisition system. However, this system is not targeted to the construction of expert systems, and it does not deal with the issues which are essential to our approach. The differences from other knowledge acquisition systems, MORE (Kahn, Nowlan & McDermott, 1985), MOLE (Eshelman, Ehret, McDermott & Tan, 1987) and SALT (Marcus, 1987), are more striking since these systems do not use classification problem-solving and presuppose domain knowledge.

This paper is organized as follows, In part 1, consisting of sections 2 and 3, we briefly explain some of the ideas on which our method is based, and in part 2, consisting of sections 4, 5, 6, 7, we describe how these ideas are applied to the construction of expert systems. Finally, we end with an appendix containing a brief session with SNOWY.

2. Background

For the last few years, we have developed a model of comprehension of elementary scientific texts or expository texts, and a program which embodies the model, called SNOWY, (Gomez, 1985; Gomez & Segami, 1989*a*,*b*). The comprehender of our model is a reader who is unfamiliar with most of the concepts which he/she is reading about. When SNOWY starts reading a scientific paragraph its domain knowledge consists only of a bare hierarchy of concepts. As SNOWY reads, new concepts are integrated in the right place in the hierarchy and new relationships about already known concepts are learned. For instance, after reading the passage

below with no knowledge of hormones and insulin, SNOWY acquires the concepts *hormones, insulin*, and *level of sugar in the blood.*

> All hormones are chemicals produced by animals. Hormones control the activities of cells. Insulin is a hormone produced by the pancreas. Insulin controls the level of sugar in the blood by making cells take in sugar.

SNOWY does not only acquire those concepts, but it also builds a classification hierarchy. For instance, the concept *insulin* is classified as a subconcept of *hormone produced by the pancreas*; this, as a subconcept of *hormones*; this, in turn, as a subconcept of *chemical produced by animals,* and, finally, this last one as a subconcept of *chemicals.* In contrast to knowledge-intensive models of comprehension which understand by framing the input in rich pre-built structures, SNOWY understands by using *formation* rules which build conceptual structures from the logical form of sentences. These conceptual structures are then passed to a *recognizer* algorithm which checks if those concepts exist in long-term memory (LTM). Finally, those concepts which the recognizer algorithm fails to recognize are passed to an integration algorithm which decides where to integrate those concepts in LTM. We have called these three phases concept formation, concept recognition and concept integration.

Although SNOWY is a knowledge-lenient model of comprehension, this does not mean that SNOWY does not possess any knowledge when it starts reading a text. Besides the linguistic knowledge which allows SNOWY to parse and disambiguate sentences (Gomez, 1988), SNOWY consists of a set of categories organized in a hierarchy, a set of primitive relationships for action and descriptive verbs, which

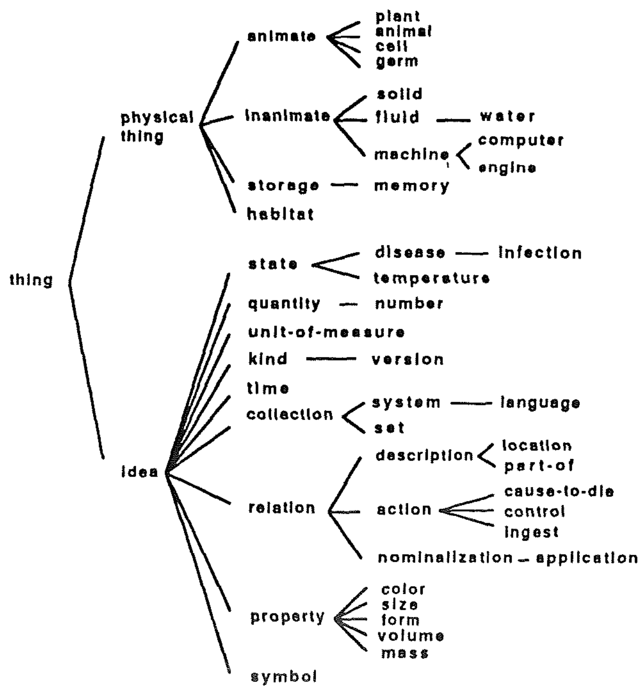


FIGURE 1. Partial content of the initial hierarchy of concepts in LTM.

are used to build the logical form of the sentence, and a set of formation rules which build the LTM conceptual structures from the logical form of the sentences. Figure 1 contains a sample of the *a priori* hierarchy, divided into physical things and ideas. The portion of the hierarchy which is shown in Figure 1 is strictly hierarchical. However, when the system begins to acquire new concepts, it builds a "tangled" hierarchy as described in the sections below. As new sentences are typed, the system builds the representation of the new concepts and relationships and inserts them into the hierarchy at the appropriate places. For example, for the sentence "germs are animates", the system adds the following to LTM:

(germ(is-a(animate)))
(animate(classes-of(germ)))

Before describing how to build these hierarchies, we need to explain briefly the knowledge representation structures used by our system. Three kinds of representation structures may be part of LTM: *object-structures, action-structures,* and *event-structures.*

An object-structure contains knowledge about physical or abstract objects and has slots called conceptual relationships or predicates. Two kinds of conceptual relationships can be distinguished: those which describe objects and those that attribute actions to physical things. This distinction corresponds to the distinction between descriptive and action verbs in natural language (Gomez, 1982). "Hormones are chemicals" and "all centipedes are slim and have many legs" are descriptive sentences, while "antibiotics kill germs" and "cells take in sugar" express actions performed by physical things. Descriptive attributes are also represented following the frame notation; for instance, the knowledge in the sentence about centipedes is represented as:

centipede
  (form(slim(q1(all))))
  (has-body-part(leg(q1(l(all))(q2(many))))

The slot q1 contains the quantifier of the concept described by the object-structure, and the slot *q*2 contains the quantifier of the second argument of the relation. Conceptual relationships denoting actions are represented in the *object-structure* as:

relation (a1) (if the relation is monadic)

If the relationship takes two arguments or more it is represented as:

(relation (concept1 a1))

where *concept1* is the second argument of the relationship. If the same relationship is true of two or more concepts, as is the case of the relationship underlying the sentence "Penguins live in the sea and in the land", the relationship is represented in the *object-structure* as:

(relation (concept1 a1) (concept2 2a)...(concepti ai))

The terms *a1, a2, ... ai* stand for the names of the conceptual relationships. The structure which represents the conceptual relationship itself is called an *action-structure*. For example, the *action-structure representing* the conceptual relationship

in the sentence "antibiotics kill germs" looks like:

a1
(args (antibiotics) (germs)
(pr (cause-to-die))
(q1 (?))
(q2 (?))

The slot *args* contains the arguments of the relationship. If the relationship is monadic, the slot *args* will contain one concept. If the relationship is diadic (as in this case), the slot *args* contains two concepts, and so on. The first concept in the slot *args* is always the actor of the action verb and the second concept is the object. The slot *pr* contains the relationship. The slot *q1* contains the quantifier of the first argument, and the slot *q2* has the quantifier of the second argument of the relationship. If the relationship is triadic, there will be a *q3* slot for the quantifier of the third argument. The representation of "antibiotics kill germs" will be done by using the following *object-structures* and *action-structure*:

antibiotics	germs
(cause-to-die (germs a1))	(cause-to-die:by (antibiotics a1))

a1
(args (antibiotics) (germs))
(pr (cause-to-die))
(how (?))
(q1 (?))
(q2 (?))

Every concept which fills a case (actor, object, destination, source, etc.) in the sentence is indexed independently. In the above example, both the actor, *antibiotics,* and the object, *germs*, are indexed using an *object-structure.* Note how the concepts *germs* and *antibiotics* point to the same *action-structure, a1,* so that the question "what kills germs?" can be answered. The question mark in the slots *q1 and q2* means that the sentence does not specify the quantifiers. If the sentence had said "all antibiotics kill germs", then the content of the slot *q1* would have been *all.* If the sentence had said "most antibiotics kill germs", the slot *q1* would contain *most*.

From a representation point of view, *event-structures are like action-structures.* However, they are used to represent relationships which are arguments of other relationships and, which can not be true if they are disconnected from the relationships in which they are embedded. For instance, it can not be concluded that "most people are happy" from the sentence "money causes most people to be

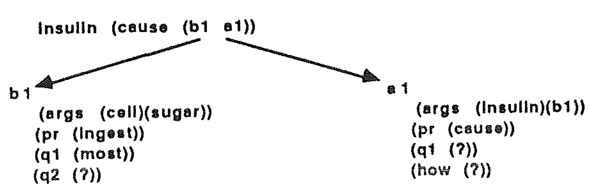


FIGURE 2. The representation of "insulin causes most cells to take in sugar".

happy". The sentence "insulin causes most cells to take in sugar", depicted in Figure 2, is represented by using three *object-structures* (insulin, cell, sugar), an *action-structure* (*a1*), and an *event-structure* (*b1*). (Gomez, 1986).

3. The detection and construction of classification hierarchies

The importance of explicit classification has been widely recognized and described by cognitive scientists working in reading. In explicit classification, the classes are clearly introduced and defined as in:

> There are two kinds of mammals: terrestrial mammals and aquatic mammals. Aquatic mammals include the carnivorous and the omnivorous. The carnivorous includes the seals and walruses. The omnivorous includes the whales and etc.

In contrast to this type of classification, we will show that if a program is to acquire knowledge by reading texts it needs to classify in many instances in which classification is not apparent *prima facie.* Underlying classification is a more basic phenomenon than that which can be observed in other forms of classification, such as generalization-based classification or conceptual clustering. Furthermore, it is essential for constructing a problem solver from an English description of the task by a human expert, since the system will detect the classification hierarchies even when the expert does not see them. In the next sections, we will show that classification hierarchies underly two frequently used sentential structures: restrictive relative clauses and existentially quantified sentences.

3.1. LEARNING CONCEPTS DENOTED BY RESTRICTIVE RELATIVE CLAUSES

A first aspect of knowledge acquisition is posed by the learning of concepts denoted by noun groups and restrictive relative clauses, e.g. "birds with long legs", hormones produced by the pancreas", "programming languages which run in PCs". The acquisition of these concepts involves their appropriate integration in the hierarchy, which in turn requires their classification by determining their parent and children nodes. For instance, consider the passage:

> Birds with long legs live in swamps. They have a long bill to extract food from under water.

This text introduces the concept *bird with long legs* for which a name is not given. However, a unique name in memory needs to be created for that concept so that additional knowledge about this concept can be learned by storing this knowledge under the same node, and so that this node can be linked to the other nodes in the hierarchy. The acquisition of the concept *bird with long legs* entails its classification so that it can be integrated appropriately in the hierarchy of concepts. If the system fails to classify the concept, it will not be able to answer such questions as "are there birds with long bills?" or "do birds with long legs have feathers?"

Restrictive relative clauses and complex noun groups are represented in LTM via *object-structures* with a dummy name (a gensym), and their representation is characterized by the presence of a *characteristic-features* slot (written *cf*), whose purpose is to identify the concept by describing the features that are characteristic of the concept. For example, when building the representation of the sentence "insulin is a hormone produced by the pancreas," the concept *hormones produced by the*

pancreas, a subconcept of the concept *hormones,* is created with the following representation:

(x1 (cf (is-a (hormone)) (make:by (pancreas (q2 (?))))))
(hormones (classes-of (x1)))

Within the *characteristic-features* slot there is always at least an is-a slot, which indicates the parent node of the concept being learned. In this example, the concept *x1* has been created. *x1* stands for a subconcept of the concept *hormones,* as indicated by the is-a slot, and is further characterized by referring to those hormones made by the pancreas. The acquisition of the concept *x1* is completed by creating the slot classes-of in the concept *hormones,* and by filling it with *x1*. Had the slot classes-of existed in the concept *hormones,* then *x1* would be added to it. The newly-created concept is identified by the content of the *characteristic-features* slot and not by the dummy name *x1*. Thus, if the question "what do hormones produced by the pancreas cause?" is asked, the system recognizes that the concept *hormones produced by the pancreas* was previously created and accesses the *object-structure x1*. (The slot *q2* contains "?" because it is unknown if every pancreas makes hormones).

The acquisition of concepts described by complex noun groups is done in a similar manner. The representation of *bird with long legs* proceeds by creating the concept *long leg* first, and then the concept *bird with long legs*. Both concepts are represented in Figure 3. This definition creates the concept *x1*, a subconcept of the concept *leg*. The concept *bird with long legs* is represented by *x2*. The *characteristic-features* slot in *x2* says that *x2* is a subconcept of *bird* characterized by having long legs. Because the noun group does not specify how many long legs *x1* has, the quantifier slot *q2*, whose scope is *x1*, has a question mark in it. If the system has knowledge about birds, it fills this slot; otherwise, it will remain with a question mark until the system finds out more about birds.

3.2. FORMATION OR LEARNING RULES

The mechanism we have described for acquiring concepts denoted by restrictive relative clauses and noun groups is based on *formation or learning rules.* These are very general rules anchored in the verbal primitives, and in some instances in lexical items. In most cases these rules use knowledge in LTM and in the structure built by the parser.

The formation rules which form the concepts *x1* and *x2*, which have been explained above, are part of the formation of noun groups. For instance, the antecedent of the rule which is fired in the example above is:

> The head noun has the feature animate and is followed by the preposition "with", and the object of the preposition has the feature body-part.

x1 (cf(is-a(leg))(size(long)))	x2 (cf(is-a(bird)(body-part(x1 (q2(?)))))
leg *classes-of(x1)*	bird classes-of(x2)

FIGURE 3. The representation of "bird with long legs".

Since the object of the preposition is also a noun group, the formation rules for the noun group are activated recursively to form the concept *long leg*. In this case, the antecedent of the rule activated is:

The head noun has the feature body-part and its modifier has the feature size.

The latter rule will form the concept *x1* and will pass it to the former rule, which will form the concept *x2*.

This has been a very simple example to illustrate basic formation rules. In the next section, we explain how formation or learning rules are used to learn more complex classification hierarchies.

3.3 THE ACQUISITION OF CLASSIFICATION HIERARCHIES

As we have seen, a classification hierarchy may be indicated explicitly, as in the sentence "antibiotics are chemicals" or it may be stated in a more subtle way as in the phrase "hormones produced by the pancreas". There is yet a more complex way of expressing a complex hierarchy of concepts like those illustrated in the passage below:

(1) Not all animals which live in the sea are fish. Some are mammals.

These two sentences introduce three concepts: *animals which live in the sea, sea fish* and *sea mammal.* The last two concepts are subconcepts of the first one.

The learning of these concepts and the hierarchy which they form is done by means of a learning rule attached to *not all* and by one of the learning rules anchored in *are.* The first sentence is parsed into:

(subj (not all animals) rela (g00001) verb (are) pred (fish))
g00001
(subj (animals) verb (live) prep (in the sea))

The learning rule which builds the representation for restrictive relative clauses is activated. This rule activates the formation rule for the verb *live.* The structure built is:

x1	animal
cf(is-a(animal) (habitat (sea)))	classes-of (x1)

The name of the concept which has been created replaces the concept modified by the relative clause in the main sentence, resulting in:

(subj (not all x1) verb (are) pred (fish))

The formation of the main sentence proceeds by activating the learning rules for *are.* The following rule fires:

If the subject is under the scope of an existential quantifier or the negation of an universal quantifier then
a. create a concept with a dummy name, say x2, and with a characteristic-features slot containing the slots is-a (subj) and is-a (pred)
b. create another concept with a dummy name, say x3, and with a characteristic-features slot containing the slots is-a (subj) and is-a (?).

The terms *pred* and *subj* in the rules are variables containing the subject and the predicate of the sentence, respectively. This rule when applied to (1) gives:

x2	x3
(cf (is-a (x1) (fish)))	(cf (is-a (x1) (?)))

The application of the rule for restrictive relative clauses and the one for *are* has resulted in the creation of three concepts: the concept *x1*, which corresponds to *sea animal,* the concept *x2*, which corresponds to *sea fish,* and *x3*, which has been only partially specified.

Adverbs such as *almost, only,* and the negation of universal quantifiers, e.g. *not all,* introduce links which connect the sentence where they appear to the next sentence. In order to solve these links, these lexical terms have attached to them formation rules. These rules are activated as part of the formation of the sentence which immediately follows the one which starts with *not all,* etc. These formation rules resolve the anaphoric reference of *some* in the second sentence and fill in the question mark in the is-a slot of the concept *x3* above. The concepts and the hierarchy formed for the passage *Not all animals which live in the sea are fish. Some are mammals* are found in Figure 4. The formation rules for *not all* do not only deal with this example, but also with examples such as:

Not all animals which live in the sea are fish. Whales are mammals.

In this case, the formation rule checks LTM to see if whales live in the sea. If so, it will replace the question mark in the concept *x3* above with *mammal*, and create *whale* as a subconcept of *x3*. However, even if there is no knowledge about *whales* in LTM, the system infers from the structure of the sentence that *whales* are sea animals.

3.4. THE ACQUISITION OF CONCEPTS DENOTED BY EXISTENTIALLY QUANTIFIED SENTENCES

The acquisition of concepts introduced by sentences starting with an existential quantifier requires the construction of a classification hierarchy in the same manner as we have done for relative clauses. The sentence "some mammals live in the sea" introduces the concept *mammal which live in the sea.* However, the representation in Figure 5 is inadequate for the job. The problem with this representation becomes obvious if the next sentence is "these mammals are intelligent". Since the structure above has not created the subconcept *mammal which live the sea* there is no node in

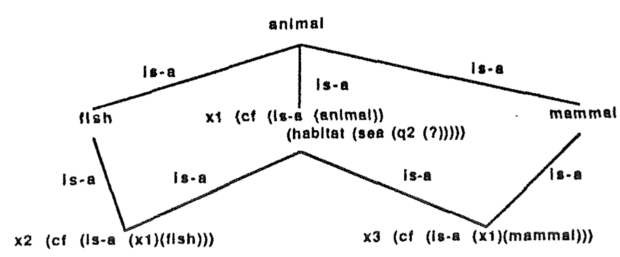


FIGURE 4. The hierarchy constructed for "Not all animals which live in the sea are fish. Some are mammals".

```
mammal                          sea
  habitat (sea a1)                habitat: by (mammal a1)

                          a1
                          args(mammal,sea)
                          pr (habitat)
                          q1 (some)
                          q2 (?)
```

FIGURE 5. Inadequate representation of "some mammals live in the sea".

the hierarchy to integrate the knowledge that they are intelligent. The representation of the sentence "some mammals live in the sea" is given below:

x1 (cf(is-a(mammal)) (habitat (sea(q2(?)))))

x1 represents the concept *animals which live in the sea.* The concept *mammal* will contain the slot "classes-of (*x1*)". The knowledge expressed by the sentence "these mammals are intelligent" can now be integrated under the concept *x1*.

The role of classification is one of the most pervasive aspects in the structure of expository prose. As a final example consider the text below:

> Some birds are unusual because they cannot fly. The emu, the rea, and the ostrich are such birds. The emu is an Australian bird. The rea lives in South America. The ostrich lives in Africa and runs very fast.
>
> Some birds are unusual because they form a partnership with an animal. The oxpecker eats insect from the rhinoceros. The honey guide will lead a badger to a bees' hive.

The authors found this text in a 6th grade comprehension exercise to teach the notion of classification to children. The children were asked to write the names of birds on the dotted lines which follow the text. This text illustrates clearly that only through classification can the concepts and relationships in the text be learned.

4. The application of these ideas to the automatic construction of expert systems

We will show in this section how these ideas can be applied to the construction of an expert system. The task which we will study is that of building a consultation system. We will use an example similar to that discussed in Boose and Bradshaw, 1987, the selection of a programming language.

The expert is instructed to present his/her subject in a top down fashion describing first the most general concepts and then refining them until reaching the most concrete concepts which will form the tip nodes in the hierarchy. The paragraph below is just one way of how this description may be accomplished. Many

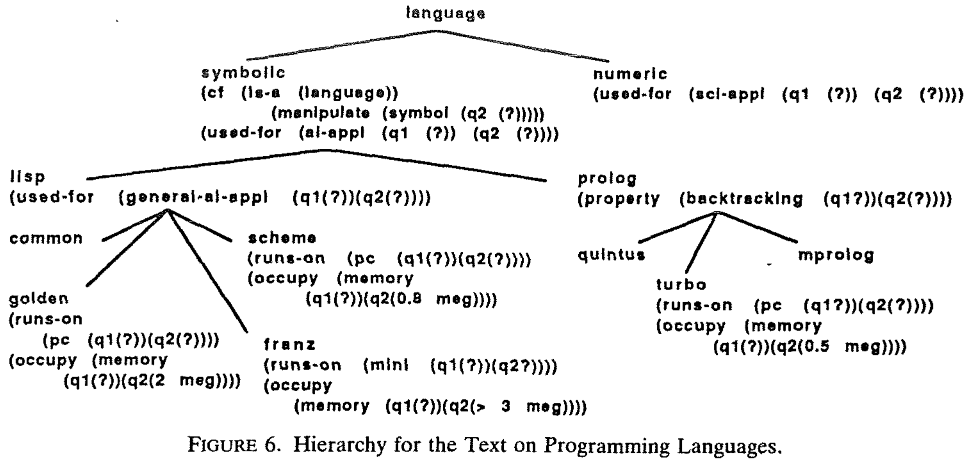


FIGURE 6. Hierarchy for the Text on Programming Languages.

others are possible. After reading this paragraph, SNOWY builds the hierarchy of concepts shown in Figure 6.

> There are several kinds of programming languages. Programming languages which manipulate symbols, called symbolic languages, are used for AI applications, while numeric languages are used for scientific applications. There are two main symbolic languages: Lisp and Prolog. Lisp is used for general AI applications, whereas Prolog is used when backtracking is needed. Common Lisp, Franz, Scheme and Golden Hill are Lisp languages. Franz runs in minicomputers and takes more than 3 megabytes of memory. Scheme is a dialect of Lisp for PCs. It takes 0·8 megabytes of memory. Golden-Hill is also a version of Lisp for microcomputers, but it requires 2 megabytes of memory. There are several versions of Prolog, such as Quintus, Turbo and Mprolog. Of these, Turbo runs on PC's, and requires only 0·5 megabytes of memory. Numeric languages. . . .

The system proceeds by parsing the sentences. In the current implementation, the system can parse any sentence containing unknown nouns and adjectives, but verbs must be known to the parser for a successful parse to occur. If the parser fails then the system asks the user for a definition of those words which are unknown to the system. First of all, the system asks the user if the word in question is a verb (the user can check that by consulting a dictionary). If the word is a verb the system obtains the syntactical usage of the verb by activating an interface which allows users without a formal knowledge of English grammar to add new verbs to the parser. The system can make sense of many sentences containing words unknown to it. We postpone the discussion of how this is done until the next section. We briefly explain some of the formation rules used to construct the hierarchy in Figure 6. The first sentence does not create any structure. From the second sentence, the system creates the concepts *symbolic language* and *numeric language,* according to the techniques which we have explained for the formation of concepts denoted by restrictive clauses and noun groups. The *characteristic-features* slot of the concept *symbolic language* contains the conceptual relationship "manipulate symbols". Note that from this moment on, the meaning of *symbolic language* is a programming language which manipulates symbols. The formation of "symbolic languages are

used for AI applications", is integrated by creating the descriptive relationship "used-for" (ai-application)" under the concept *symbolic language.* The next sentence creates two subconcepts of *symbolic language*: Lisp and Prolog. The next sentence differentiates the two languages. The formation rule which builds the LTM representation of "Prolog is used when backtracking is needed" constructs the following structure:

Prolog
  (property(backtracking))

This is a general rule which forms all logical forms which fall under the schema:

<somebody> <use> <obj1> when/if <somebody> <need> <obj2>

into the structure:

obj1(property(obj2))

The rule is using the inference that if something, say x, is used when something else, say y, is needed then y is a property of x. This is necessary if requests such as "I need a symbolic language with backtracking" are to be answered by the consultation system.

5. Defining unknown words

One of the critical issues in the design of the system is how many concepts in the user's input need to be previously known by the system in order for it to understand the text and solve problems. If the knowledge acquistion system is going to be domain independent, then the system can not have *a priori* knowledge about concepts which are domain specific. For instance, the concept of "backtracking" in the sample text can not be part of the *a priori* concepts. However, if the knowledge acquisition system is going to be domain-dependent, then one can provide it with a rich set of domain dependent concepts. Since our goal is to build a domain independent knowledge acquisition system, the problem of unknown concepts becomes critical. The sample text can be processed by the present implementation without knowing anything about the following concepts: "programming languages", "symbols", "AI", "numeric languages", "backtracking", "Lisp languages", "megabyte" and "Lisp", "Prolog", etc. If one asks the system "what is backtracking", this will reply by saying "I do not know what it is, but I know that it is a property of Prolog". The system's reply to a question is "I do not know" if the concept in the question is not linked through the *is-a* relationship to the concepts in the *a priori* hierarchy. For classification problem solving, the system does not need to have a deep understanding of what backtracking is. In any case, the decision of determining if a concept should be fully known to the system rests on the user.

In some cases, however, it is essential to know the meaning of some terms, because without knowing them the meaning of verbs, prepositions and attachement of prepositions can not be determined. If one does not know the meaning of "evolution" in the sentence "Peter read a book on evolution in 5 days", the attachment of "in 5 days" can not be decided. Let us discuss these issues in the context of acquiring the other concepts in the text. The sentence in which the

concept "megabyte" appears for the first time is "Franz ... takes 3 megabytes of memory". Let us assume that the system knows that memory is something where information is stored. The problem here is that when the system is going to form the final knowledge representation from the logical form of the sentence, it cannot determine the meaning of "of" because it does not know the meaning of "megabyte". In these cases, the system displays the possible meanings of "megabyte" from the context and asks the user to select the most appropriate one. In this case, the system selects these meanings from the formation rules which are used to determine the meaning of "of". (The actual display for this case is illustrated in the appendix.) For instance, the system will display:

Is megabyte a kind of memory?
Is megabyte made of memory?
Is megabyte a unit of measure of memory?
....
....

In other cases, the user may decide to define a concept which is unknown to the system by using one of the three methods explained briefly below.

• *Classification*: The best way to define a word is by providing the upper-concept of the concept denoted by the word to be defined. For instance, the word AI can be defined by saying "AI is a computer science". An inappropriate definition will be to say something like "AI tries to make computers intelligent", since this does not place AI in the hierarchy of concepts which forms the basis of SNOWY's understanding process. Likewise, the definition which the system is expecting of *backtracking* is not a long and intricate one, which probably will be irrelevant to the expert system being built, but just something like "backtracking is a programming technique" which places *backtracking* in the hierarchy of concepts. We are extending the *a priori* hierarchy of categories to include a large set of concepts so that it will be easy to place new concepts in the hierarchy by classifying them.
• *Part-of definitions*: Some words may be defined by using a *part-of* relationship. For instance, the word *cpu* or *computer memory* can be defined by saying thay they are part of a computer.
• *Functional definitions*: We are studying short functional definitions of concept from which the system can infer the *part-of* relationship plus some other information. For instance, the concept *computer memory* above can be defined as: "memory is something to store information into a computer". From this, the system can infer that *computer memory* is a physical thing, part of a computer and is used to store information. The reason why this method of definition has not been incorporated in the system, yet, is because it opens the door for a user to give a definition too complex to be understood.

6. Problem-solving

After SNOWY has read the paragraph of section 4, a user interested in a consultation about programming languages may, at this point, run a consultation test. For example, the user may say "I am interested in a language that runs on PC's

and is used for ai applications". Given this inquiry, the task of the system is to find the languages that have the features "runs on PC's" and "used-for ai applications". The class of languages that have these features is represented as:

x1 (cf(is-a(language)) (runs-on(pc)) (used for (ai-appl)))

If this concept were present in LTM, the question would be answered by simply retrieving its children. Since this is not the case, one way to find the answer to the question is to *classify* the concept *x1* ("languages that run on PC's and are used for *ai*") within the current hierarchy of concepts. This task is done by a *classifier algorithm,* which determines where in the current hierarchy a new concept, such as *x1*, would be located if it existed in LTM, that is, the *classifier* determines which of the concepts currently in LTM are the parents of *x1* and which ones are its children. We briefly illustrate the steps taken by the *classifier,* using the concept *x1* as an example. See Gomez and Segami (1989*a*) for a detailed description of the algorithms in this section and a comparison with other classifier algorithms. Given the representation structure of *x1*, the concept within its *is-a* slot determines what part of LTM needs to be considered by the *classifier.* In this case, only the concept *language* or its children can be the parents or the children of *x1*. The *classifier* performs a traversal of selected nodes under the concept *language.* The concept in each visited node is compared with the concept *x1*. Depending on whether one is a subclass of the other, or there is no hierarchical relationship between them, the *classifier* decides which node to visit next. The actual comparison of concepts is performed by the *compare-classes* algorithm, which is based on the following idea. A concept *x2* is a subclass of a concept *x1*, if each characteristic-feature of *x1* is true of each entity in the class *x2*. In our example, concept *x1* is first compared with the concept *symbolic languages.* No hierarchical relationship is found between these two concepts. However, one of the *characteristic-features* of *x1*, "used-for *ai* applications", is found to be true of all the descendants of the concept *symbolic languages,* so that, when the nodes under *symbolic languages* are examined, only one more feature needs to be verified: "runs on PC's". After doing this, it is found that *scheme, golden,* and *Turbo* run on PC's. Then, the *classifier* returns the list "[(language) (scheme golden turbo)]" to indicate that the concept "languages that run on PC's and are used for *ai*" is a child of the concept *language* and that it has the children *scheme, golden,* and *Turbo.* The system, then, would reply to the user as follows: "the following languages run on PC's and are used for *ai*: scheme, golden, and turbo". If the user decides to narrow down his/her choices, he/she may either specify additional features, or formulate questions. For example, the following interaction may now take place:

user: the language must run in a computer with less than 1 megabyte of memory
system: scheme and turbo require less than 1 megabyte of memory
user: what are the differences between scheme and turbo?
system: scheme is a lisp dialect used for general ai applications while turbo is a prolog dialect used for backtracking

The first sentence typed by the user simply adds one more requirement to the desired programming language. Since we have an initial set of solutions, the new requirement needs to be verified for the members of this set only. This reduces the

possible solutions to two. The second sentence typed by the user is a question, which requires that a comparison be made between the features of *Scheme* and *Turbo*. This is accomplished by traversing the hierarchy upwards, from the nodes corresponding to *Scheme* and *Turbo*. The traversal stops when the two paths cross each other.

The *classifier* is the component that allows SNOWY to maintain a correct organization of its memory. Each new statement entered by the expert may cause the insertion of a new concept in the hierarchy, or it may trigger a reorganization of the concepts in LTM. Let us suppose, for example, that the expert continues entering information to the system, by typing the following:

> Some languages are procedural. All procedural symbolic languages are widely available.

The sentence "some languages are procedural" is integrated as explained in section 3.4. SNOWY's long-term memory configuration for the two sentences is depicted in Figure 7. If the expert now types the statement: "Lisp is a procedural language", the concepts in LTM will be reorganized as shown in Figure 8. Thus, in an inquiry of the form "a language that runs on PC's is used for ai applications, and is widely available", only the Lisp dialects will be considered to be possible solutions.

As one can see the problem-solving method which we have illustrated uses only categorical knowledge (Szolovits & Pauker, 1978) and not probabilistic knowledge. It is our belief that probabilistic reasoning plays a minor role in most problem-solving situations if memory organization and indexing form part of the problem-solving method. In any case, certain aspects of probabilistic reasoning need to be included in our system. Experts use sentences like this "Lisp is easier to learn than Pascal which in turn is easier to learn than Cobol". In cases like this, we need to ask

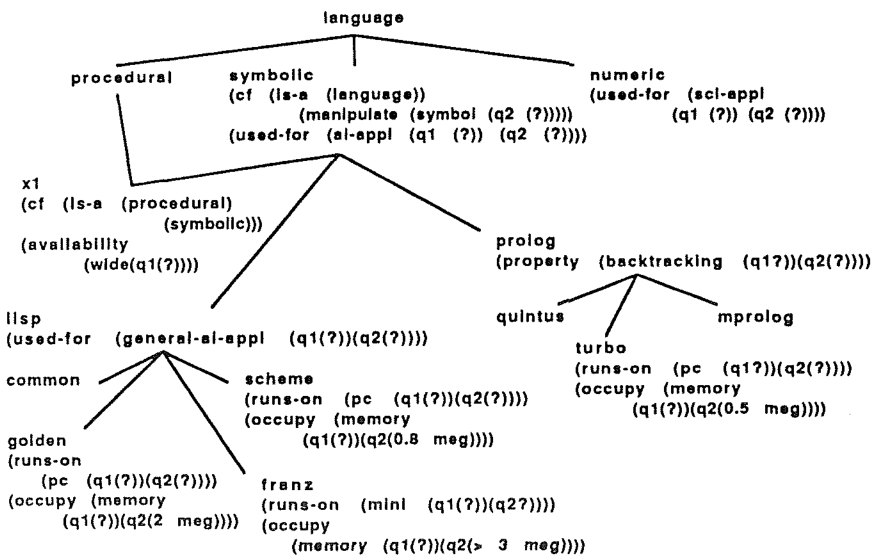


FIGURE 7. Organization of LTM after processing the passage "Some languages are procedural. All procedural symbolic languages are widely available".

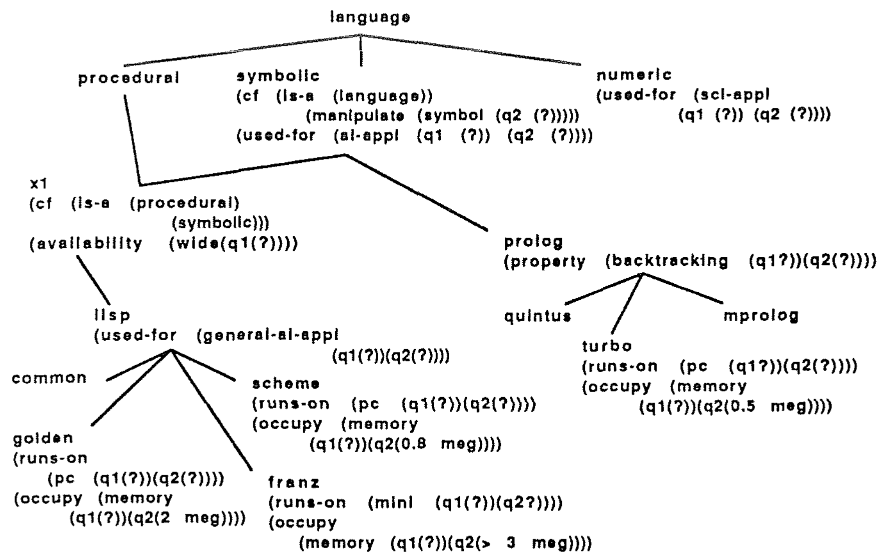


FIGURE 8. Reorganization of the concepts in LTM after processing the sentence "Lisp is a procedural language.

the expert to rate Lisp, Pascal and Cobol with respect to the predicate *learn.* There are many ways in which this rating can take place and AQUINAS offers several ways which we plan to adopt in our system. Let us consider for the sake of the discussion that the expert produces the following rating: (Lisp (learn ·9)) (Pascal (learn ·7)) (Cobol (learn (·4))). Then, the phrase "a symbolic language that is easy to learn" in the user's request "I am interested in a symbolic language which runs in a PC and is easy to learn", will be translated into a numeric value which may be greater than 5 according to the rating provided by the user for that predicate.

7. Discussion

In this paper, we have shown how a consultation system can be built from a natural language description by a human expert. It is our belief that the ideas and techniques described here can be used for the construction of expert systems for tasks which can be modelled using classification problem-solving. These tasks do not have to deal with consultation systems. They may be about diagnosis, design, etc. The relevance of the ideas presented in this paper go beyond the knowledge acquisition area, since we have introduced a problem-solver which has striking differences from existing rule-based systems. In our system, organization and reorganization of memory, comprehension and problem solving are integrated processes. As a result, two important aspects of expert system technology become almost trivial: the updating of the knowledge base and interacting with the expert system. However, our approach is not without limitations. Let us first discuss the theoretical limitations. Although we believe that a large class of problems and domains can be handled using our system, there are many domains which are beyond reach of our method. The reason for that is that there are many aspects of

the real world which can not be represented using the classification hierarchies we have presented in this paper. For instance, the functioning of complex systems such as the circulatory system, or the respiratory system can not be adequately grasped using our present representation techniques. We are working in knowledge representation structures which will represent these systems from a point of view which will be adequate for comprehension and problem-solving (Gomez, 1990).

Although the capabilities of the system have grown considerably since the submission of this paper, there are limitations to our present implementation. First of all, no expert has yet used our program to build a consultation system. All the consultation examples have been entered by graduate students or by the implementers of the sytem. Our system is a prototype which we can "demo" at any time, but that has not been used by any real user. This is clearly in contrast with AQUINAS which, according to the authors, has been used to build a variety of consultation systems. We are going to start to experiment with some real users to assess the real world capabilities of our system. Since SNOWY is able to acquire new concepts and new words, it can augment its lexicon by interacting with a user. The parser (Gomez, 1988) being used by SNOWY is a modular algorithm based on the notion of *syntactical usage of a word,* which allows a naive user without knowledge of English grammar to increase its syntactical coverage. We are writing an interface to our parser which will allow a user to add new verbs not presently in the parser. Yet, we are aware that there is a qualitative jump from testing a prototype by people who have certain familiarity with it to a real world system used by naive users.

We have presented a system and distinguished it from existing knowledge acquisition sytems. Yet, our approach can be integrated with present knowledge acquisition tools and vice versa, techniques in other systems can be incorporated in our system. We think that, in future developments of our system for real users, the elicitation techniques developed for ETS and AQUINAS can be very useful. In particular, we are thinking that in those cases in which the English of certain users becomes too hard to handle, SNOWY may fall back on elicitation techniques. Likewise, domain dependent knowledge acquisition tools can benefit from a system which is able to build classification hierarchies from natural language text.

References

BOOSE, J. (1984). Personal construct theory and the transfer of human expertise. *Proceedings of the National Conference on Artificial Intelligence, Austin, Texas.*

BOOSE, J. & BRADSHAW, J. (1987). Expertise transfer and complex problems: using AQUINAS as a knowledge acquisition workbench for expert systems. *International Journal of Man–Machine Studies,* **26,** 3–28.

CHARNIAK, E. & MCDERMOTT, D. (1984). *Introduction to Artificial Intelligence.* Reading, MA: Addison-Wesley.

CLANCEY, W. J. (1985). Heuristic classification, *Artificial Intelligence* **27,** 289–350.

ESHELMAN, L., EHRET, D., MCDERMOTT, J. & TAN, M. (1987). MOLE: a tenacious knowledge acquisition tool. *International Journal of Man–Machine Studies,* **26,** 41–54.

GOMEZ, F. (1982). Towards a theory of comprehension of declarative texts. In *Proceedings of the 20th Meeting of the Association of Computational Linguistics, Toronto, Canada,* 36–43.

Gomez, F. (1985). A model of comprehension of elementary scientific texts. In *Proceedings of the Workshop on Theoretical Approaches to Natural Language Understanding, Halifax Nova Scotia,* 70–81.

Gomez, F. (1986). *Knowledge Representation for Understanding Expository Texts.* Technical Report CS-TR-86, Department of Computer Science, University of Central Florida, Orlando, Florida.

Gomez, F. (1988). WUP: A parser based on word usage, in *Proceedings of the 1988 IEEE Conference on Computers and Communication, Scottsdale, Arizona,* 445–449.

Gomez, F. & Chandrasekaran, B. (1984). Knowledge organization, and distribution for medical diagnosis. In W. Clancey & E. Shortliffe, Eds. *Readings in Medical Artificial Intelligence.* Reading, MA: Addison-Wesley.

Gomez, F. & Segami, C. (1989*a*). The recognition and classification of concepts in understanding scientific texts. *Journal of Experimental and Theoretical Artificial Intelligence,* **1,** 51–77.

Gomez, F. & Segami, C. (1989*b*). Classification-based inferences in retrieving information from a database of scientific facts. In *Proceedings of the Fifth IEEE Conference on Artificial Intelligence Applications, Miami, Florida.*

Gomez, F. (1990). *Representing Biological and Physical Systems as Temporal Event Hierarchies.* Technical Report CS-TR-90-02, Department of Computer Science, University of Central Florida, Orlando, Florida.

Haas, N. & Hendrix, G. (1980). An approach to acquiring and applying knowledge. *Proceedings of the National Conference on Artificial Intelligence, Stanford, CA,* 235–239.

Kahn, G., Nowlan, S. & McDermott, J. (1985). MORE: an intelligent knowledge acquisition tool. *Proceedings of Ninth International Conference on Artificial Intelligence, Los Angeles, California.*

Marcus, M. (1987). Taking backtracking with a grain of SALT. *International Journal of Man–Machine Studies,* **26,** 383–398.

Szolovits, P. & Pauker, S. (1987). Categorical and probabilistic reasoning in medical diagnosis. *Artificial Intelligence,* **11,** 115–144.

Appendix I

SAMPLE SESSION WITH SNOWY

NOTES: (1) SNOWY's knowledge before reading the sentences below, is described by the hierarchy of Figure 1.

(2) Lines preceded by ">>>" correspond to the user's input to SNOWY. Except for the remarks, other lines correspond to SNOWY's response.

(3) Symbols of the form *dmyxx* denote *object-structures* having a *characteristic-features* slot.

>>>what is a language?
:::language is-a system

>>>what languages are there?
:::I don't know

REMARKS: As indicated in the hierarchy of Figure 1, initially SNOWY knows that the concept *language* is a subconcept of *system*. Since SNOWY does not have any other knowledge about *language*, the question "what is a language?" is answered "language is-a system", and the response is "I don't know" when the question is "what languages are there?"

>>>there are two kinds of programming languages.
>>>programming languages which manipulate symbols, called symbolic languages, are used for ai applications, while numeric languages are used for scientific applications.
>>>lisp and prolog are symbolic languages.
>>>fortran is a numeric language.
>>>lisp is used for general ai applications.
>>>prolog is used when backtracking is needed.
>>>franz and scheme are lisp languages.

REMARKS: The user has typed several sentences which SNOWY has parsed, formed and integrated in LTM. Concepts such as *kind, symbol,* plus all the verbal primitives are part of the initial hierarchy of concepts. Other concepts such as *programming languages which manipulate symbols, scientific application, backtracking,* etc., are formed and inserted in LTM.

>>>franz runs on minicomputers and takes 3 megabytes of memory.
:::unable to determine meaning of verb: *runs-on*
:::because concept *minicomputer* is unknown
:::please indicate which of the following is the most appropriate statement:

:::(1) minicomputer is a kind of machine
:::(2) minicomputer is a kind of computer

>>>2.
:::integrating *minicomputer* as a subconcept of *computer*

::unable to form prepositional phrase: *3 megabytes of memory*
:::because concept *megabyte* is unknown
:::please indicate which of the following is the most appropriate statement:

:::(1) megabyte is a kind of memory
:::megabyte is made of memory
:::megabyte is a unit of measure of memory
:::megabyte is a part of memory

>>>3.
:::integrating *megabyte* as a subconcept of *unit-of-measure*

REMARKS: When SNOWY reads this sentence, it finds that it can not determine the verbal primitive corresponding to *runs on*, due to the fact that the concept *minicomputer* is unknown. Using, as a guide, the formation rules associated with verb *runs*, SNOWY asks for clues that may lead to the disambiguation of the verb. In this case, it determines that the primitive corresponding to *runs on* is *drive.* A similar situation occurs when SNOWY examines the noun phrase "3 megabytes of memory". In this case, SNOWY obtains from the user the fact that *megabyte* is a *unit of measure.*

>>>scheme runs on pcs and takes 1 megabyte of memory.

:::unknown concept found: *pc*
:::assuming it is a kind of *machine*
:::is this a valid assumption?

```
>>>yes, pcs are microcomputers.
:::in the following hierarchy, which of the categories is the most appropriate for
*microcomputer*?

machine
  computer
    minicomputer
  engine

>>>computer.
:::integrating *microcomputer* as a subconcept of *computer*
```

REMARKS: The verb phrase *runs on* is found for a second time in this sentence and, as in the first case, SNOWY is unable to determine the corresponding primitive due to the fact that the concept *pc* is unknown. In this case, though, SNOWY assumes that *runs on* has the same meaning as before, and therefore, that "pcs are machines", and requests the user to confirm this. The user answers affirmatively and goes on to specify that "pcs are microcomputers". Since *microcomputer* is an unknown concept, SNOWY displays the current hierarchy under the concept *machine* and asks the user to select the most adequate category for the new concept. In any case, the user may override SNOWY's assumptions by providing a negative answer or by indicating that none of the choices given are appropriate. In future implementations, much of this interaction will be done through a graphic interface with the help of a mouse.

```
>>>common-lisp and golden-hill are versions of lisp.
>>>golden-hill runs on microcomputers.
>>>golden-hill requires 2 megabytes of memory.
>>>quintus and turbo are versions of prolog.
>>>turbo runs on pcs and requires 2 megabytes of memory.
>>>(showc language)

language
  programming-language
    dmy0 ((is-a(programming-language)) (manipulate(symbol)))
      lisp
        common-lisp
        golden-hill
        franz
        scheme
      prolog
        quintus
        turbo
    numeric-language
      fortran
```

REMARKS: The command "showc" displays the hierarchy of concepts that has been built under the concept *language*. Programming languages are categorized as *programming languages which manipulate symbols*, called *symbolic-languages*, and *numeric-languages*. Once SNOWY has acquired some knowledge from a domain expert, a user of the system may interact with it, in this case, to select a programming language. The following questions illustrate this.

>>>what kinds of programming languages are there?
:::symbolic-language is-a programming-language
:::numeric-language is-a programming-language

>>>what symbolic languages are there?
:::lisp is-a <programming-language manipulate symbol>
:::prolog is-a <programming-language manipulate symbol>
:::common-lisp is-a lisp
:::golden-hill is-a lisp
:::franz is-a lisp
:::scheme is-a lisp
:::quintus is-a prolog
:::turbo is-a prolog

REMARKS: In response to this question, SNOWY displays all languages that are symbolic languages. The concept *symbolic-language* is stored in LTM under the dummy name *dmy0*, and is defined by its *characteristic-features* "is-a programming-language" and "manipulate symbols." Thus, a question like "what languages manipulate symbols?" would have the same answer.

>>>which of these languages run on pcs?
:::scheme drive pc
:::turbo drive pc

>>>how much memory is required by scheme?
:::scheme occupy 1 (megabyte) memory

>>>is scheme used for ai applications?
:::yes

>>>is turbo used for ai applications?
:::yes

>>>what is the difference between scheme and turbo?
:::scheme used-for general-ai-application
:::turbo property backtracking

Appendix II

PARSING AND FORMATION OF CONCEPTS

In this appendix we show the parser and formation phase outputs for sample sentences. Symbols of the form *gxx* denote structures built by the parser,

corresponding to sub-clauses within the main clause, *@xxx* denote *action-structures*, while the symbols *dmyxx* denote *object-structures* having a *characteristic-features* slot.

Given the sentence

programming languages which manipulate symbols are used for ai applications

the parser returns the structures:

```
g76
(subj(somebody) verb (are used) obj ((adj programming) (noun languages))
  rela g77 prep (for ((noun ai) (noun applications)))
g77
(subj ((adj programming) (noun languages)) verb (manipulate) obj ((noun
  symbols)))
```

These structures are passed on to the formation phase, which builds the representation structures below.

```
((dmy0(cf(is-a(programming-language)) (manipulate (symbol (q2(?)))))
  (manipulate (symbol ($more (@a1))))
  (used-for (ai-application (q1 (?)) (q2 (?)))))
(@a1 (args (dmy0) (symbol)) (pr (manipulate)) (q1 (all)) (q2 (?)))
(symbol (manipulate%by (dmy0 ($more (@ai)))))
(ai-application (used-for%by (dmy0)) (is-a (application))))
```

The concept *programming languages which manipulate symbols* is represented by the *object-structure dmy0*, which contains a *characteristic-features* slot. Some of the quantifier slots contain a question mark to indicate that the quantifier is unknown. These structures become the input to the Integration Phase, which links the new concepts to the concepts currently in LTM by defining the *is-a* and *classes-of* slots.

As a second example, consider the sentence

scheme runs on pcs and takes 1 megabyte of memory

The parser output is

```
g434
(subj ((noun scheme)) verb (runs-on) obj ((nouns pcs)) conj (and) link g344)
g344
(subj (subj-of-prev) verb (takes) obj ((num 1) (noun megabyte)) prep (of((noun
  memory))))
```

which the Formation Phase transforms to

```
((scheme (drive (pc ($more (@a4))))
  (occupy (memory (q2 (1 (megabyte))) (q1 (?)))))
(pc (drive%by (scheme ($more (@a4)))))
(@a4 (args (scheme) (pc)) (pr (drive)) (q1 (?)) (q2 (?)))
(memory (occupy-by (scheme (q1 (1 (megabyte))) (q2 (?))))))
```

Appendix E

To appear in IEEE Transactions on Systems, Man and Cybernetics

Classification-Based Reasoning

Fernando Gomez
Department of Computer Science
University of Central Florida
Orlando, Fl 32816
gomez@ucf.csnet

Carlos Segami
Department of Mathematics and Computer Science
Barry University
Miami Shores, Fl 33161

Abstract

A representation formalism for N-ary relations, quantification and definition of concepts is described. Three types of conditions are associated with concepts: (1) necessary and sufficient properties, (2) contigent properties and (3) necessary properties. It is also explained how complex chains of inferences can be accomplished by representing existentially quantified sentences, and concepts denoted by restrictive relative clauses as classification hierarchies. The representation structures which make possible the inferences are explained first, followed by the reasoning algorithms which draw the inferences from the knowledge structures. All the ideas explained in this paper have been implemented and are part of the information retrieval component of SNOWY, a program which understands scientific paragraphs. The paper ends with an appendix containing a brief session with the program.

This is a modified and highly extended version of the paper "Classification-Based Inferences in Retrieving Information from a Database of Scientific Facts" which appeared in *Proceedings of the Fifth IEEE Conference on Artificial Intelligence Applications*, Miami, Florida, 1989.

This work was supported by NASA, Kennedy Space Center, under grant No. NAG-10-0058.

1. Introduction

For the last few years, we have been working on a model of comprehension of scientific texts [7]. We have also built a computer program, called SNOWY, which embodies this model [9]. SNOWY is a knowledge-lenient model of comprehension which reads texts with minimal knowledge or no knowledge at all about their contents. This aspect differentiates SNOWY, which builds conceptual representations from scratch by using formation, or learning, rules, from knowledge-intensive models of comprehension based on the notions of plans [21], MOPs [18] or TAUs [5]. Also, our work differs from recent research on discourse which has focused on dialogue or conversation [13,1,16,12,20].

The system has the following components: parsing, forming, recognizing, finding explanations, and integrating concepts into long-term memory (LTM). The Formation Phase builds from the output of the parser [8] the LTM knowledge representation structures used by the system. The Recognition Phase determines which concepts and relations built by the Formation Phase are known to the system. The Explanation Phase tries to understand the explanatory links connecting sentences (e.g. Antibiotics work against infections. They kill the germs which cause them.), and the Integration Phase integrates into LTM the concepts and relations which the Recognition Phase fails to recognize.

In [10], we explained briefly how this paradigm of comprehension differs from knowledge-intensive models of comprehension and, then, we went on to explaining in depth the recognition, classification and integration algorithms. In this paper, we describe the knowledge representation structures and the inference algorithms which allow SNOWY (1) to answer questions posed by users. In particular, in this paper we pay special attention to the representation of quantification, N-ary relations and the definition of concepts. Likewise, special emphasis has been put in showing that the algorithms described in this paper derive valid inferences.

The key concept on which SNOWY relies to draw inferences is the notion of *classification*. Consider the following paragraph below:

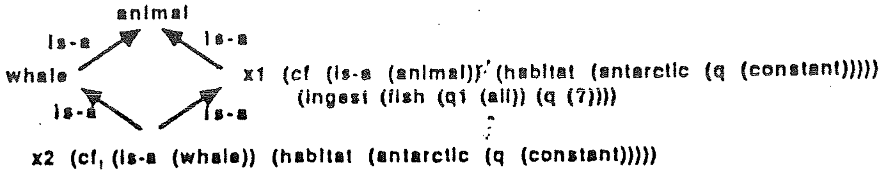
All whales are animals. All animals which live in the Antarctic eat fish. Some whales live in the Antarctic.

These sentences can be represented in Predicate Calculus as (the existential quantifier is represented as Vx, and the universal quantifier as (x)):

1. (x) (Wale(x) --> Animal(x))
2. (x)(Animal(x) -> (Live(x,Antarctic)) -> Vy(Fish(y) and Eat(x,y))))
3. Vz(Whale(z) and Live(z,Antarctic))

Suppose that given the above paragraph, the following question is asked: Do whales eat fish?" The Predicate Calculus representation of the question is VxVy(Whale(x) and Fish(y) and Eat(x,y)). Using resolution to answer the question is beyond the point, because the database of facts does not consist of just the facts listed above, and resolution does not offer any indexing mechanism to access the right chunk of knowledge.

Yet, the inference can be drawn relatively easily if restrictive relative clauses and existentially quantified sentences are represented as classification hierarchies, and a classification algorithm is invoked to integrate the newly created concepts in the right place in long-term memory (LTM). The representation of restrictive relative clauses and existentially quantified sentences as classification hierarchies is one of the major aspects which distinguishes our approach from KL-ONE [2,19]. Other differences are the representation of N-ary relations, quantification and the association of three types of properties with a given concept: (1) clusters of necessary and sufficient conditions, (2) necessary conditions and (3) contingent conditions. The inference algorithms which we will discuss in the next sections are a direct consequence of this representation. In our example, the clause "all animals which live in the Antarctic" will be represented as a subclass of *animals* distinguished by the fact that they live in the Antarctic; the sentence "whales are animals" will be integrated as "whale -- is-a --> animal;" and, when the sentence "some whales live in the Antarctic" is read, a subconcept *of whale* is created, namely, the concept denoted by "those whales which live in the Antarctic." Then, when the concept *whales which live in the Antarctic* is integrated in LTM, the classification algorithm determines that it must be integrated as a subconcept of the concepts *whale* and *animals which live in the Antarctic*, as illustrated in the diagram below. (The meaning of the terms used in the diagram is explained in Sections 2 and 3.) With this hierarchy of



concepts in LTM, the answer to the question "do whales eat fish" can be obtained by examining the concept *whale,* then descending to the concept *whales which live in the Antarctic* and ascending to the concept *animals which live in the Antarctic.* (The precise way in which this done is explained in Section 5, Table 2.)

In the example we have just seen the answer is obtained by examining the concepts already in LTM. However, there is a type of inference which takes place when a question asks information about a concept which does not exist in LTM. For instance, suppose that the facts referred by the following sentences are represented in LTM:

All birds are animals. All animals which live in the Antarctic swim.

The question is "do birds which live in the Antarctic swim?" In this case, the answer to the question can not be found in LTM, because the concept *birds which live in the Antarctic* does not exist in LTM. The only way in which we or the system could have formed the concept *birds which live in the Antarctic* is if we had been told that "birds live in the Antarctic." However, the answer can be obtained by activating the *Classifier* algorithm and asking where the concept *birds which live in the Antarctic* would be placed in LTM, if it existed there. The *Classifier* will link the concept *bird which live in the Antarctic* to the upper-concept *animal which live in the Antarctic,* and the answer to "do birds which live in the Antarctic swim?" can be obtained by inheritance.

In order to set up the stage to properly discuss the reasoning algorithms, we need to explain the knowledge representation structures used by SNOWY. This is a necessary component of every paper dealing with SNOWY's algorithms since these depend on the knowledge representation structures. However, in this paper we have provided First Order Predicate Calculus (FOPC) formulae to make clear the meaning of the representation structures. Sections 2 and 3 deal with the knowledge representation structures, the representation of restrictive clauses and existentially

quantified sentences. In section 4, we explain is-a inheritance in SNOWY. In sections 5 and 6, we explain the reasoning techniques based on our representation structures. In Section 7, we explain briefly the types of questions handled by SNOWY. In section 8, we give our conclusions. Finally, we have included an appendix containing a brief session with SNOWY.

2. Representation of Concepts in LTM

This section explains the representation of concepts in LTM, so that the reader can follow the subsequent sections. In order for SNOWY to understand a text, it needs to start with at least a minimum set of concepts, which categorizes the world into states, actions, collections, etc. These concepts are organized in a hierarchy (Figure 1). Generic concepts in the hierarchy are linked through *is-a* and *classes-of* slots. All of these concepts denote non-empty sets of individuals. Indi-

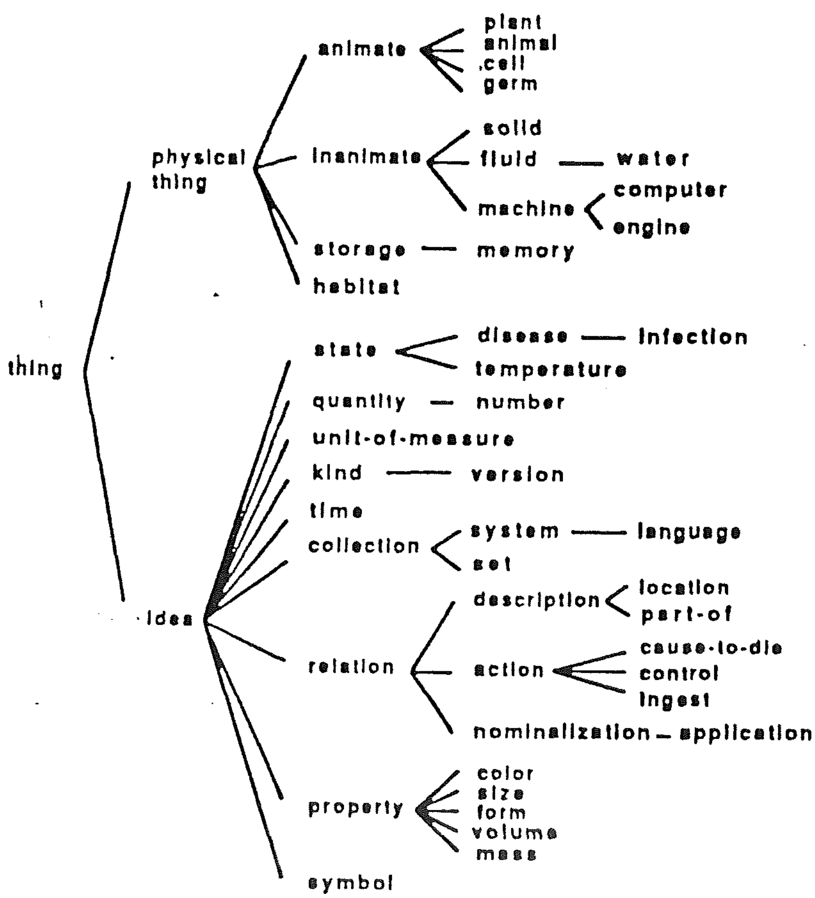


Figure 1. Partial Contents of the Initial Hierarchy of Concepts in LTM

vidual members of a class, such as Peter or Fido, are linked to their parent concepts through *instance-of* and *instances* slots. As SNOWY starts reading a text and begins adding new concepts to the hierarchy, this becomes a "tangled" hierarchy. New concepts are recognized and integrated in this *a priori* hierarchy. The concepts in the hierarchy are represented using three kinds of representation structures: *object-structures, action-structures,* and *event-structures.* Table 1, below, describes the syntax of *object-structures* and *action-structures.* The syntax of *event-structures* is the same as that of *action-structures*, except that the slot "(instance-of (event))" is always present in *event-structures.*

An *object-structure* contains knowledge about physical or abstract objects. As shown in Table 1, an *object-structure* consists of the object's name, an optional *cf* slot (explained in detail in Section 3), and one or more slots called conceptual relations or attributes. Two kinds of conceptual relations can be distinguished: those which describe objects and those that attribute actions to physical things. This distinction corresponds to the distinction between descriptive and action verbs in natural language [6]. "Hormones are chemicals" and "all centipedes are slim and have many legs" are descriptive sentences, while "antibiotics kill germs" and "cells take in sugar" express actions performed by physical things. The knowledge in the sentence about centipedes is represented as:

```
(centipede
  (form (slim (q1 (all))))
  (has-body-part (leg (q1 (all)) (q (many)))))
```

This example corresponds to grammar rules 1, 2, 4, and 5 for *object-structures*, in Table 1. The nesting of the quantifiers in the structure is from left to right. Thus, the meaning of the second relation in the *object-structure* for *centipede* is (Since every concept in LTM denotes a non-empty set of individuals, the FOPC formula below should be preceded/followed by "∀x(Centipede(x)) and ∀y(Leg(y))." We postpone the discussion of this until the next section.):

$$(x)\ (Centipede(x) \rightarrow \forall y(Leg(y) \text{ and } Has\text{-}body\text{-}part(x,y)))$$

The slot *q1* contains the quantifier of the concept described by the *object-structure*, and the slots *q* contain the quantifier of the other arguments of the relation. Note that because FOPC does not have a quantifier for "many" we have represented "many" as an existential quantifier. (These

Object Structures

```
<object-struct>  ::= ( <object-name> [ <cf-slot> ]  { <slot> }+ )

<slot>           ::= <action-slot>  |  <desc-slot>

<action-slot>    ::= ( <action-relation> { ( [ <argument> ]  { <action-struct-name> }+ ) }+ )

<desc-slot>      ::= ( <descriptive-relation>  { <desc-arg> }+ )

<desc-arg>       ::= ( <object-name> [(q1 (<quantifier>))] [(q (<quantifier>))]
                         { ( <relation> { ( <object-name> (q (<quantifier>))) }+ ) } )

<cf-slot>        ::= ( cf (is-a {(object-name)}+) { <desc-slot> } { <cf-action> } )

<cf-action>      ::= ( <action-relation> (( [<object-name> (q (<quantifier>))]  { <case-slot> } ) } )
```

Action Structures

```
<action-struct>  ::= ( <action-struct-name> <args-slot> <relation-slot>  { <case-slot> }+ )

<args-slot>      ::= ( args { ( <argument> ) }+ )

<relation-slot>  ::= ( pr ( <action-relation> ) )

<case-slot>      ::= ( <case-name> ( <object-name>  ( q ( <quantifier> ))))      |
                         ( <case-name> ( <event-struct-name> ( q (some))))

<argument>       ::= <object-name>  |  <event-struct-name>

<action-struct-name> ::= a1 | a2 | .....

<event-struct-name>  ::= e1 | e2 | .....

<object-name>    ::= animate | mammal | .....

<action-relation> ::= ingest | make | .....

<quantifier>     ::= all | some | constant | many | most | few | .....

<case-name>      ::= source | destination | instrument | .....
```

Table 1. The Syntax of Object-Structures and Action-Structures

NOTES: (1) The notation {A} denotes zero or more occurrences of the string A, {A}+ denotes one or more occurrences of A, while [A] denotes zero or one occurrence of A. (2) The syntax for *event-structures* is the same as *action-structures*, except that the slot "(instance-of (event))" must always be present in *event-structures*.

issues are discussed in detail in the section below.) Conceptual relations denoting actions are represented in the *object-structure* as:

(concept1 (action-relation (a1))),

if the relation takes no object, while relations that do take an object are represented as:

(concept1 (action-relation (concept2 a2)))

where *concept2* is the object of the relation. If the same relation is true of two or more concepts, it is represented as:

(concept1 (action-relation (concept2 a2) (concept3 a3) ... (concepti ai))

Thus, the relation underlying the sentence "Germs attack animals and plants" is represented in the *object-structure* of *germ* as:

(germ (attack (animal a2) (plant a3)))

This example corresponds to grammar rules 1, 2, and 3 for *object-structures*, in Table 1. The terms *a1, a2, ... ai* stand for the names of the conceptual relations. The structure which represents the conceptual relation itself is called an *action-structure*. For example, the *action-structure* representing the conceptual relation in the sentence "bacteria cause diseases" is as follows:

```
(a1                                  (inst a1 cause)
  (args (bacteria) (disease))        (actor a1 bacteria)
  (pr (cause))                       (object a1 disease)
  (actor (bacteria (q (?))))         (quantifier-subject a1 unknown)
  (object (disease (q (?)))))        (quantifier-object a1 unknown)
```

Of these two representations, we will use the one on the left side. The one on the right side is a slot-assertion notation [4]. The slot *args* contains the arguments of the relation. If the relation is monadic, *args* will contain one concept. If the relation is diadic (as in this case), *args* contains two concepts, and so on. The slot *pr* contains the relation. The slots *q* contain the quantifier of the argument where the slot is embedded. A question mark in a quantifier slot means that the contents of *q* may be a universal quantifier or an existential quantifier. When an argument of a relation is an individual member of a class (i.e., Peter, Fido) its quantifier slot will contain the value *constant*, to indicate that it is a member of a class (see for example *Antarctic* in the diagram of page 3). Note that necessary conditions are associated with a concept since the value of a

```
(bacteria (cause (disease a1)))          (a1
                                           (args (bacteria)(disease))
                                           (pr (cause))
(disease (cause:by (bacteria a1)))         (actor (bacteria (q (?))))
                                           (object (disease (q (?)))))
```

Figure 2. The Representation of "bacteria cause diseases"

```
(blood (transport (oxygen a1)))          (a1
                                           (args (blood)(oxygen)(cell))
(oxygen (transport:by (blood a1)))         (pr (transport))
                                           (actor (blood (q (?))))
(cell (destination-of (a1)))               (object (oxygen (q (?))))
                                           (destination (cell (q (?)))))
```

Figure 3. The Representation of "Blood transports oxygen to cells"

---

quantifier can be "all" in an *action-structure.*

The representation of "bacteria cause diseases" is done by using the *object-structures* and *action-structure* which are displayed in Figure 2. Every concept which fills a case (actor, object, destination, source, etc.) in the sentence is indexed independently, with pointers to the *action-structure* (conceptual relation) of which they are part. In the above example, both the actor, *bacteria,* and the object, *diseases,* are indexed using an *object-structure.* Note how the concepts *diseases* and *bacteria* point to the same *action-structure.* Similarly, Figure 3 shows the representation of the relation underlying the sentence "Blood transports oxygen to cells." This is a relation with three arguments, *blood, oxygen,* and *cells.* Each of them has its own *object-structure* in LTM, and each one of these structures points to the *action-structure* representing the given relation.

The Meaning of Action-Structures

Consider the sentence "all people like dogs." The *action-structure* for the representation of this sentence is:

```
(a1
  (args (human)(dog))
  (pr (like))
  (actor (human (q (all))))
  (object (dog (q (some)))))
```

The meaning of *action-structures* similar to the above, for different values of the quantifiers, is given below. (We denote the universal quantifier with (x) and the existential quantifier with Vx.) Also, we assume that the implication connective has the lowest precedence followed by "or" and "and."

1. (a1
(args (human)(dog))
(pr (like))
(actor (human (q (all))))
(object (dog (q (all)))))

(x)(y) (Human(x) and Dog(y) --> Like(x,y)) and Vx(Human(x)) and Vy(Dog(y))

2. (a1
(args (human)(dog))
(pr (like))
(actor (human (q (all))))
(object (dog (q (some)))))

(x) (Human(x) --> Vy (Dog(y) and Like(x,y))) and Vx(Human(x)) and Vy(Dog(y))

3. (a1
(args (human)(dog))
(pr (like))
(actor (human (q (some))))
(object (dog (q (all)))))

Vx(Human(x) and (y) (Dog(y) --> Like(x,y))) and Vy(Dog(y))

4. (a1
(args (human)(dog))
(pr (like))
(actor (human (q (some))))
(object (dog (q (some)))))

VxVy(Human(x) and Dog(y) and Like(x,y))

The FOPC formulas for structures (1), (2) and (3) are followed by Vx(Human(x)) and Vy(Dog(y)) because in our representation every concept in LTM denotes a non-empty set of individuals. The formula (x)(y) (Human(x) and Dog(y) --> Like(x,y)) is not a correct representation of structure (1), since this formula is true even if there are no humans or dogs. Therefore, our representation of "all Greeks liked unicorns" includes *unicorns* as entities in our ontology, since these putative entities will be under the scope of an existential quantifier [18]. Then, how will the algorithm know that unicorns did not live? This could be done by creating a category in our *a priory* hierarchy of entities which do not have physical existence, and properly indexed under them will be such things as

unicorns, fairies, etc. So, if SNOWY were asked the question: Were there unicorns? the answer will be "Of course, and all Greeks liked them," but they do not have a physical existence.

From here on, we will omit the existentially quantified predicates (at the end of the formulas) which indicate that concepts represented by *object-structures* denote non-empty sets of individuals.

As one can see, the nesting of the quantifiers is from left to right. Hence, the *action-structures* above can not represent the following two statements: "There is a dog which is loved by everybody" and "Every dog is loved by somebody," which correspond to the two FOPC formulas below, respectively.

5. Vy(Dog(y) and (x) (Human(x) --> Like(x,y)))

6. (y) (Dog(y) --> Vx(Human(x) and Like(x,y)))

In fact, formulas (3), (4) and (5) will be not be represented using an *action-structure*. Since they correspond to existentially quantified sentences, they are are represented as classification hierarchies, as indicated in the introduction and explained in detail in the next section. Formula (6) above is represented by using the inverse relation of *like*, namely *like:by*, in the following way:

```
(human
  (like (dog a1)))

(dog
  (like:by (human a1)))
```

```
(a1
  (args (dog) (human))
  (pr (like:by))
  (actor (dog (q (all))))
  (object (human (q (some)))))
```

The meaning of *a1* is:

(y)(Dog(y) --> Vx(Human(x) and Like:by(y,x)))

Consider the following structure in LTM:

```
(a1
  (args (whale)(fish))
  (pr (ingest))
  (actor (whale (q (?))))
  (object (fish (q (all)))))
```

The question mark in the quantifier slot of concept *whale* means that the content of *q* may be a universal quantifier or an existential quantifier. Then, the structure above may mean:

(x)(y) (Whale(x) and Fish(y) --> Eat(x,y))

or,

$\forall x(\text{Whale}(x) \text{ and } (y) (\text{Fish}(y) \rightarrow \text{Eat}(x,y)))$

If in a chain of inferences the reasoning algorithm comes across an argument (in a relation) which is quantified with a question mark, it interprets the question mark as an existential quantifier. This is consistent with not amplifying the premises or conclusion in an inference. Other quantifiers such as "many," "most," "few," etc. are also taken as existential quantifiers in drawing deductive inferences. Yet, these quantifiers play an important role in the inductive learning algorithms which we are presently constructing for SNOWY.

In summary, the semantics of our representation structures is obtained by translating them into FOPC formulae (the nesting of quantifiers is from left to right). Any interpretation that makes true the FOPC formula makes also true our corresponding knowledge representation structure.

The Representation Of Embedded Relations

A relation which is an argument of another relation is represented using an *event-structure*. In the sentence, "insulin causes cells to take in sugar" the conceptual relation underlying the clause "cells take in sugar" is represented using an *event-structure*. From a representational point of view, *event-structures* are like *action-structures*, except that they contain the slot "instance-of (event)." The important thing to note here is that relations represented by *event-structures* can not be concluded to be true separate from the *action-structures* in which they are embedded. For instance, from "alcohol causes most people to do foolish things," it can not be concluded that "most people do foolish things." The representation of "insulin causes most cells to take in sugar" (see Figure 4) is done by using an *object-structure*, an *action-structure* and an *event-structure* (assuming that "insulin" is universally quantified). In all of our examples so far, the object of an action has been a concept denoting entities. However, in the sentence "insulin causes most cells to take in sugar" the object is an event. Thus, in Figure 4, the *object-structure* is (insulin (cause(e1 a1)), where the object of the relation *cause* is the relation represented by the *event-structure e1*. Therefore, the *action-structure a1*, which represents the relation *cause*, has *e1* in its *args* and *object* slots. The question "what does insulin cause" is answered by examining the *action-structure a1* and retrieving the object, which in this case is the structure e1: "most cells ingest

sugar." The meaning of the representation in Figure 4 is: (Note that the meaning of "most" is lost in the FOPC formula.)

(x)(Insulin(x) --> VyVz(Cell(y) and Sugar(z) and Cause(x,Ingest(y,z))))

Prior to discussing how quantifiers in the embedding relation and the embedded relation relate one to another, consider the sentences below:

(1) Everyone wants to own a house.
(2) Everyone wants everyone to own a house.

From (1), it can be concluded that Peter wants to own a house, but not that Peter wants Mary to own a house. However, both inferences can be concluded from (2). The representation of (1) and (2) are depicted in Figures 5 and 6, respectively. In the representation of (1), the universal quantifier "all" has an index, *x*, in the *action-structure* and in the *event-structure*. This is so, to indicate that the concept "human" in the *event-structure* is the same as the one in the *action-structure*. The FOPC formulas for the structures in Figures 4 and 5 are, respectively:

(3) (x)(Human(x) --> Vy(House(y) and Want(x, Own(x,y))))

```
(insulin (cause (e1 a1)))

(e1
  (instance-of (event))
  (args (cell)(sugar))
  (pr (ingest))
  (actor (cell (q (most))))
  (object (sugar (q (?)))))

(a1
  (args (insulin)(e1))
  (pr (cause))
  (actor (insulin (q (all))))
  (object (e1 (q (some))))
  (instrument (?)))
```

Figure 4. The Representation of "insulin causes most cells to take in sugar"

```
(a1
  (args (human)(e1))
  (pr (want))
  (actor (human (q (allx))))
  (object (e1 (q (some)))))

(e1
  (instance-of (event))
  (args (human)(house))
  (pr (own))
  (actor (human (q (allx))))
  (object (house (q (some)))))
```

Figure 5. The Representation of "everyone wants to own a house"

```
(a1                              (e1
  (args (human)(e1))               (instance-of (event))
  (pr (want))                      (args (human)(house))
  (actor (human (q (all))))        (pr (own))
  (object (e1 (q (some)))))        (actor (human (q (all))))
                                   (object (house (q (some)))))
```

Figure 6. The Representation of "everyone wants everyone to own a house"

---

(4) (x)(Human(x) --> Vy(House(y) and (z)(Human(z) --> Want(x, Own(z,y)))))

An indexed quantifier may appear in an *action-structure* without any *event-structures* embedded in it, as in the sentence "everyone likes himself/herself" whose representation is:

```
(a1
  (args (human)(human))
  (pr (like))
  (actor (human (q (allx))))
  (object (human (q (allx)))))
```

3. Representing Concepts Denoted by Restrictive Relative Clauses and Existentially Quantified Sentences as Classification Hierarchies

The representation of concepts denoted by restrictive relative clauses and other restrictive qualifiers such as prepositional phrases involves their appropriate integration in the hierarchy. Concepts such as *hormones produced by the body*, *germs which enter the body* or "cars with fuel injection" are described by complex descriptions. Yet, these concepts need to have a unique name in memory so that knowledge about them can be integrated under the same node. These concepts are represented in LTM via an *object-structure* with a dummy name (a gensym), and their representation is characterized by the presence of a *characteristic-features* slot whose purpose is to describe the features that are characteristic of the concept. For example, when building the representation of the sentence, "insulin is a hormone produced by the pancreas," the concept, *hormones produced by the pancreas*, a subconcept of the concept *hormones*, is created with the following representation:

```
(x1 (cf (is-a (hormone)) (make:by (pancreas (q (?))))))
(hormones (classes-of (x1)))
```

This concept is identified by the contents of the *characteristic-features* slot and not by the dummy

name x1. Thus, if a question like "what do hormones produced by the pancreas cause?" is asked, the system recognizes that the concept "hormones produced by the pancreas" was previously created and accesses the *object-structure* x1. (The slot *q* contains "?" because it is unknown if every pancreas makes hormones.)

The meaning of the *cf* slot above can be expressed by the following FOPC (we are assuming that the value of the question mark is "some").

(x) (X1(x) iff (Hormone(x) and Vy(Pancreas(y) and Make:by (x,y)))

From a formal point of view, the *cf* slot can be viewed as playing the same role as a meaning postulate [3]. The *cf* slot identifies a concept by providing necessary and sufficient conditions. At least one of these conditions must a parent concept.

Complex noun groups such as *sea mammals* and nouns modified by prepositional phrases such *birds with long legs* are represented in a similar form. The concept *sea mammal* is represented as:

(x1 (cf (is-a (mammal)) (habitat (sea (q (?))))))
(mammal (classes-of (x1)))

How the category "habitat (sea)" is computed does not concern us in this paper.

The Representation of Existentially Quantified Sentences

Sentences which begin with an existential quantifier such as "some mammals live in the sea" require a representation similar to the one we have proposed for restrictive relative clauses. Consider the sentence "some mammals live in the sea." If an unique node in memory is not created for the concept *mammals which live in the sea* then subsequent knowledge about this concept could not be integrated. Creating two object structures "mammal" and "sea" linked to an *action-structure* containing the relation *habitat*, quantifier of the actor = some, and quantifier of the object = ?, will not do any good if the next sentence is "these animals are in great danger." The conceptual relation underlying this concept could not be stored under the node corresponding to *mammals which live in the sea*, since this concept would not exist in LTM. Hence, the representation of the sentence "some mammals live in the sea" is:

(x1 (cf (Is-a(mammal)) (habitat(sea (q (?))))))
(mammal (classes-of (x1)))

Note that because every concept in LTM denotes a non-empty set of individuals, the existence of mammals is captured in this representation. This representation of existentially quantified sentences as classification hierarchies is consistent with the *use* of such sentences in expository discourse. Constructions like "Some programming languages are used for numeric computation. Numeric programming languages ...." and many others similar to this abound in expository prose. It will take us well beyond the limits of this paper to provide further details to back up this view (see [11]).

4. Is-a Inheritance

Prior to our discussion of inheritance in SNOWY, we introduce the following definitions.

Definition 4.1

Let x and y be concepts in LTM. Concept y is said to be a "proper subclass" of x if there exists a sequence of concepts y1, y2,...,yn in LTM, such that y1=y, yn=x, and the relation "yi Is-a y(i+1)" exists in LTM for all i=1,....,n-1. Concept y is said to be a "subclass" of x if either y=x, or y is a proper subclass of x. If y is a proper subclass of x, then y is also called a "descendant" or "sub-concept" of x, and x is called an "ancestor" or "superconcept" of y.

There are two ways in which inheritance mechanisms work in our representation language: (1) by means of *is-a* relations combined with universal quantification and (2) through *characteristic-features* slots. Let us suppose that the following two sentences are read:

All animals produce chemicals called hormones. Whales are animals.

The concepts and relations underlying these sentences are integrated in LTM as follows:

```
chemical                    (a1                                  animal
  ↑ is-a                      (args (animal) (hormone))          (make (hormone a1))
hormone                       (pr (make))                          ↑ is-a
(make:by (animal a1))         (actor (animal (q (all))))         whale
                              (object (hormone (q (?)))))
```

If the question "do whales produce hormones?" is asked, the concept *whale* is examined first,

looking for the conceptual relation "whale make hormone." Because the conceptual relation is not found there, the concepts in the *is-a* slot of *whale* are examined next. Since the conceptual relation "animal make hormone" is found under the concept *animal*, the *action-structure* a1 will be searched for the contents of the quantifier *q1*. If this quantifier is found to be "all," as in this case, the answer to the question is "yes." However, if the first sentence had been "animals produce chemicals called hormones," the quantifier *q1* in the *action-structure* a1 would not be "all," and the question "do whales produce hormones?" would have been answered "I don't know." Thus, the descendants of a concept inherit only those conceptual relations in which the concept is universally quantified.

The second case is inheritance through *characteristic-features* slots. To illustrate this case, we use an example for which an *object-structure* containing a *characteristic-features* slot is built. Suppose the sentences "All hormones are chemicals produced by animals," and "Insulin is a hormone" are read. The underlying concepts and relations are integrated as:

```
              chemical
                 ↑ is-a
x1 (cf (is-a (chemical)) (make:by (animal (q (?)))))
                 ↑ is-a
              hormone
                 ↑ is-a
              insulin
```

Note that the representation of "All hormones are chemicals produced by animals" is obtained by first building the concept *chemicals produced by animals*, denoted by *x1* in the figure above, and then integrating *hormone* as a subconcept of *x1*. The concept *insulin* is then integrated as a subconcept of *hormone*. Given this, suppose the question "is insulin made by animals?" is asked. The answer will be "yes," because the conceptual relation "x1 make:by (animal)" is a characteristic feature of the class x1, and all subclasses of x1 inherit such conceptual relations. However, if the question is "is insulin made by all animals," the answer is "I don't know."

We formalize the ideas in the previous examples with the following definition.

Definition 4.2

Let "B1 r B2, B3,..., Bk" denote a conceptual relation, where B1,...,Bk denote concepts, and r denotes a relation whose arguments are B1,...,Bk. Suppose that this relation exists in LTM.

(a) Let the concept A be a proper subclass or an instance of B1.

(1) If B1 is universally quantified in the relation "B1 r B2, B3,..., Bk," or if "B1 r B2, B3,..., Bk" is a characteristic feature of B1, then concept A inherits the relation "A r B2, B3,..., Bk" from B1.

(2) If B1 has an indexed universal quantifier in the relation "B1 r B2, B3,..., Bk," or in the characteristic feature "B1 r B2, B3,..., Bk," then A inherits from B1 the relation obtained from "B1 r B2, B3,..., Bk" by replacing with A each argument that has the same indexed quantifier as B1.

(b) If a concept A is a proper subclass or an instance of Bi for some i, i=2,3,...,k, then A inherits the conceptual relation "B1 r B2,..,B(i-1), A, B(i+1),..., Bk" from Bi, if Bi is universally quantified in the conceptual relation "B1 r B2, B3,...., Bk."

Case (a.2) above, is illustrated with the following example. If "all americans want to own a house," and "all texans are americans," then the concept "texans" inherits the relation underlying "all texans want to own a house" from the concept "americans."

5. Inferring the Answer from the Classification Links in LTM

In this section we study the kinds of inferences that may be drawn based on the classification links among concepts in LTM. For example, suppose that the following facts are known:

All infections are diseases
All bacteria are germs
Bacteria cause infections

If the question "Do germs cause diseases?" is asked, the system should answer affirmatively by reasoning as follows: "bacteria are germs, bacteria cause infections, and infections are diseases, so it can be inferred that some germs cause some diseases." Similarly, from the facts:

All red cells are cells
Sugar is a chemical
Insulin causes all cells to take in sugar

we would like to infer that "insulin causes all red cells to take in some chemicals," and from the facts:

All arctic whales are whales

All arctic whales are arctic mammals
All arctic mammals feed on all arctic mollusks
All arctic clams are arctic mollusks
All arctic clams are clams

we would like to infer the fact that "some whales feed on some clams." All these inferences are based on the classification links among concepts. We introduce some criteria under which inferences of this type can be made, and show that these criteria lead to valid inferences.

Theorem 1

Let A1, A2,..., Ak and B1, B2,..., Bk be concepts in LTM, and let r be a relation. Let us assume that none of the arguments in the conceptual relation "B1 r B2, B3,..., Bk" has an indexed quantifier. Then, the conceptual relation "B1 r B2, B3,..., Bk" implies the conceptual relation "A1 r A2, A3,..., Ak" (or, the second one can be inferred from the first one), if for each i, i=1,...,k, either

(1) Ai = Bi; or,

(2) Bi is a proper subclass of Ai; or,

(3) Ai is a proper subclass of Bi and Ai inherits from Bi the relation "B1 r B2, B(i-1), Ai, B(i+1),..., Bk;" or,

(4) There exists a concept Ci in LTM which is a proper subclass of both Ai and Bi, such that Ci inherits from Bi the relation "B1 r B2, B(i-1), Ci, B(i+1),..., Bk."

Furthermore, the quantifiers of A1, A2,..., Ak, in the conceptual relation "A1 r A2, A3,..., Ak" can be determined as follows:

a) concept Aj is universally quantified if case (3) is true for Aj and Bj;

b) concept Aj is existentially quantified if case (2) or case (4) is true for Aj and Bj;

c) if case (1) is true for Aj and Bj, then Aj has the same quantifier that Bj has in the conceptual relation "B1 r B2, B3,..., Bk."

Remark We note here that theorems 1, 2 and 3 will also hold if the term "a proper subclass of" is replaced with the term "a proper subclass of or an instance of," and the quantifier *constant* is treated as an existential quantifier. Thus, these theorems can also be applied when some of the arguments of the relations are constants. We do not explicitly include this case in the theorems

for the sake of simplicity.

Proof

We first prove the following lemma.

Lemma 1 Under the hypotheses of Theorem 1, for any j, j = 1,..,k, the relation

(I) "B1 r B2,.., B(j-1), Bj, B(j+1),.., Bk"

implies the relation

(II) "B1 r B2,.., B(j-1), Aj, B(j+1),.., Bk"

where the quantifier of Aj in (II) is determined according to the criteria of Theorem 1.

Proof Four cases are possible.

(1) Aj=Bj.

In this case, clearly (I) implies (II), since they are identical, and the quantifiers of Aj and Bj are the same.

(2) Bj is a proper subclass of Aj.

To show that (I) implies (II), let us assume that relation (I) holds true, that is, that it exists in LTM. If (I) holds true and Bj is a proper subclass of Aj, then (II) is true for some entities in the class Aj, namely those in Bj. Thus, (I) implies (II) with Aj existentially quantified.

(3) Aj is a proper subclass of Bj, and Aj inherits the relation (II) from Bj.

If (I) holds true, because of inheritance (I) must hold true for all entities in the class Bj and, in particular, for all entities in class Aj, since Aj is a proper subclass of Bj. Thus, (I) implies (II), and Aj is universally quantified in (II).

(4) Cj is a concept which is a proper subclass of both Aj and Bj, and Cj inherits the relation

(III) "B1 r B2,..,B(j-1), Cj, B(j+1),.., Bk"

from Bj.

Applying case (3), above, to the conceptual relations (I) and (III), we conclude that (I) implies (III) with Cj universally quantified in (III). Applying case (2) to the conceptual relations (III) and (II), we conclude that (III) implies (II), with Aj existentially quantified in (II). Thus, (I) implies (II) with Aj

existentially quantified in (II). This completes the proof of Lemma 1.

Proof of the theorem

Suppose the conceptual relations "B1 r B2, B3,..., Bk" and "A1 r A2, A3,..., Ak" satisfy the hypotheses of the theorem. Then, applying Lemma 1 with j = 1, we conclude that the conceptual relation "B1 r B2, B3,..., Bk" implies "A1 r B2, B3,..., Bk," where A1 is quantified according to the criteria of the theorem. Next, we note that the hypotheses of Theorem 1 are satisfied by the relations "A1 r B2, B3,..., Bk" and "A1 r A2, A3,..., Ak." Applying Lemma 1 to these relations, with j = 2, we conclude that "A1 r B2, B3,..., Bk" implies "A1 r A2, B3,..., Bk." Therefore, "B1 r B2, B3,..., Bk" also implies "A1 r A2, B3,..., Bk." Repeating this argument k times, we conclude that "B1 r B2, B3,..., Bk" implies "A1 r A2, A3,..., Ak," where the quantifiers of A1, A2,..., Ak are determined by the criteria of the theorem. This concludes the proof of the theorem.

Answering Verification Questions

Let "A1 r A2, A3,..., Ak" be the schema of a verification question, (e.g. "do germs enter the body?") where each Ai is a concept and r is a relation. The answer to the question is affirmative if there exists a concept B1 in LTM whose *object-structure* contains the conceptual relation "B1 r B2, B3,..., Bk" and "B1 r B2, B3,..., Bk" implies the conceptual relation "A1 r A2, A3,..., Ak."

We now illustrate with examples some of the cases included in the criteria of theorem 1.

Example 1 Consider the following inference:

All infections are diseases
Some germs cause infections

Some germs cause some diseases

Here, the relations underlying the statements "all infections are diseases" and "some germs cause infections" are assumed to be represented in LTM. Thus, we have, A1 = B1 = germ, A2 = disease, B2 = infection, and B2 is a subclass of A2. Because condition 2 of the theorem is true for A2 and B2, A2 is existentially quantified in the inferred relation.

Example 2

All hormones are chemicals produced by animals
All insulins are hormones

All insulins are produced by animals

The representation of the first two conceptual relations above, is illustrated in the figure below. Note how the concept "chemicals produced by animals" is built, and given the dummy name *x1*. In this example, A1 = insulin, A2 = animal, B1 = x1, and B2 = animal. Thus, A1 is a subclass of B1, and A1 inherits the relation "insulin make:by animal" from B1 (*characteristic-features* slot inheritance).

```
                     chemical
                       ↑ is-a
x1 (cf (is-a (chemical)) (make:by (animal (q (?)))))
                       ↑ is-a
                     hormone
                       ↑ is-a
                     insulin
```

Example 3

All hormones are chemicals produced by all animals
All whales are animals
All insulins are hormones

All insulins are produced by all whales

The representation of the concepts and relations for the first three statements above, is illustrated as follows:

```
                                      chemical
  animal                                ↑ is-a
is-a ↑       x1 (cf (is-a (chemical) (make:by (animal (q (all)))))
  whale                                 ↑ is-a
                                      hormone
                                        ↑ is-a
                                      insulin
```

Here, A1 = insulin, A2 = whale, B1 = x1, and B2 = animal. Thus, A1 is a subclass of B1 and A1 inherits the relation "insulin make:by animal" from B1 due to the universal quantifier. Also, A2 is a subclass of B2 and it inherits the relation "x1 make:by whale" from animal. Since condition 2 of the criteria of theorem 1 holds true for A1 and B1, and A2 and B2, A1 and A2 are both universally

quantified in the inferred relation.

Example 4 This example illustrates inheritance in a "tangled hierarchy."

All sea fish are fish
All sea fish are sea animals
All sea animals live in the sea

Some fish live in the sea

Suppose that the system reads the sentence "some animals which live in the sea are fish." This sentence will be integrated as shown in the figure below. In order to answer affirmatively the question "do fish live in the sea?" we need to reach the concept *x1* (sea animal) from "fish." Hence, we need to descend to the concept *x2* (sea fish) and then ascend to *x1*. In this case, A1 = fish, B1 = x1, A2 = B2 = sea. If we take C1 = x2, then C1 is a subclass of both A1 and B1, and C1 inherits the relation "C1 habitat sea" from B1. Thus, A1 is existentially quantified in the inferred relation, and A2 has the same quantifier as B2.

```
              animal
      is-a ↗        ↖ is-a
fish          x1 (cf (is-a (animal)) (habitat (sea (q (?)))))
      is-a ↖        ↗ is-a
        x2 (cf (is-a (x1) (fish)))
```

Example 5

Blood transports oxygen from the lungs to all cells
All oxygens are chemicals
All muscle cells are cells

Blood transports some chemicals from the lungs to all muscle cells

In this case, A1 = B1 = blood, A2 = chemical, B2 = oxygen, A3 = B3 = lung, A4 = muscle cell, and B4 = cell. Thus, B2 is a subclass of A2, A4 is a subclass of B4 and A4 inherits the relation "B1 r B2 B3 A4" from B4. In the inferred relation, A2 has an existential quantifier and A4 has a universal quantifier.

Theorem 1 deals with relations all of whose arguments are entities. In many instances, some arguments of a conceptual relation may be conceptual relations. For example, in the conceptual relation underlying the sentence "insulin causes all cells to take in sugar" the actor is a physical

object, while the object of the relation is a conceptual relation, i.e., the conceptual relation underlying the sentence "all cells take in sugar." From a statement like "insulin causes all cells to take in sugar" we would like to infer, for example, that "insulin causes all red cells to take in sugar." We now extend the criteria of Theorem 1 to handle the more general case when one or more arguments are conceptual relations.

Theorem 2

Consider the conceptual relations "A1 r A2,..., Ak, A(k+1),..., An" and "B1 r B2,..., Bk, B(k+1),..., Bn," where A1,...,Ak, and B1,...,Bk denote entities, A(k+1),...,An and B(k+1),...,Bn denote conceptual relations, r is a relation, and the arguments in each one of the relations B(k+1),...,Bn and A(k+1),...,An are all entities. Let us suppose further that none of the arguments in the conceptual relation "B1 r B2,..., Bk, B(k+1),..., Bn" has an indexed quantifier. Then, the conceptual relation "B1 r B2,..., Bk,..., Bn" implies the conceptual relation "A1 r A2,..., Ak,..., An," if A1,A2,...,Ak and B1,B2,...,Bk satisfy the conditions of Theorem 1, and for each i, i = k+1,...,n, the conceptual relation Bi implies the conceptual relation Ai, also according to the criteria of Theorem 1.

Proof

The proof of this theorem follows along the same lines as that of Theorem 1. We only give a brief outline of it here. Under the hypotheses of the theorem, we first show that, for any j, j=(k+1),...,n, the relation

(I) "B1 r B2,..,Bk,B(k+1),..,B(j-1),(Bj.1 rj Bj.2,..,Bj.q,..,Bj.m),B(j+1),..,Bn"

implies the relation

"B1 r B2,..,Bk,B(k+1),..,B(j-1),(Bj.1 rj Bj.2,..,Aj.q,..,Bj.m),B(j+1),..,Bn"

for any q, q=1,...,m. Since q is any integer between 1 and m, we can apply this argument m times to conclude that the relation (I), above, implies the relation

"B1 r B2,..,Bk,B(k+1),..,B(j-1),(Aj.1 rj Aj.2,..,Aj.q,..,Aj.m),B(j+1),..,Bn"

Since j is any integer between (k+1) and n, we can, in turn, apply this argument (n-k) times to conclude that the relation (I), above, implies the relation

(II) "B1 r B2,..,Bk,A(k+1),..,A(j-1),(Aj.1 rj Aj.2,..,Aj.q,..,Aj.m),A(j+1),..,An"

Finally, k applications of Lemma 1 to relation (II), allows us to conclude that relation (I) implies the relation

"A1 r A2,..,Ak,A(k+1),..,A(j-1),(Aj.1 rj Aj.2,..,Aj.q,..,Aj.m),A(j+1),..,An"

which proves the theorem.

In theorems 1 and 2 we imposed the restriction that the quantifiers in the conceptual relations not be indexed quantifiers. We would like to consider now the case in which some of the arguments of a conceptual relation do have an indexed quantifier. As we have seen, these relations correspond to sentences like "all americans want to own a house," "the human body kills germs by producing chemicals," and "all humans like themselves." We show how indexed quantifiers are dealt with in order to draw inferences based on the notion of classification.

Definition 5.1

Two indexed quantifiers are said to be the same, if they are both existential or both universal, and they have the same index (x, y, z, etc.).

Theorem 3

Let

(I) B1 r B2,..., Bk, B(k+1),..., Bn

and

(II) A1 r A2,..., Ak, A(k+1),..., An

be conceptual relations, where B1,...,Bk, and A1,...,Ak denote entities, B(k+1),...,Bn and A(k+1),...,An denote conceptual relations, r is a relation, and the arguments in each one of the relations B(k+1),...,Bn and A(k+1),...,An are all entities. Let us suppose further that concept B1 has an indexed quantifier, and let Qb = {Bu1, Bu2,..., Bug, Bv1.w1, Bv2.w2,..., Bvh.wh} be the set of arguments in the conceptual relation (I) that have the same indexed quantifier as B1 (here, the symbols Bv.w refer to arguments in the embedded relations B(k+1),...., Bn). Then, the conceptual relation (I) implies the conceptual relation (II), if conditions (a) and (b), below, are satisfied:

(a) A1 has an indexed quantifier in relation (II), and the set of arguments in (II) that have the same indexed quantifier as A1 is Qa = {Au1, Au2,..., Aug, Av1.w1, Av2.w2,..., Avh.wh}.

(b) If (I') and (II') are the relations obtained from (I) and (II), respectively, by eliminating the indices of all indexed quantifiers, then (I') implies (II') according to Theorem 2.

Furthermore, except for the indices, the quantifiers of A1, A2,..., Ak are determined according to the criteria of Theorem 1.

Proof

Again, we only outline the proof of this theorem. First, we take care of all arguments with indexed quantifiers. We note that the arguments in the set Qb are all equal to B1 and, similarly, all arguments in the set Qa are equal to A1. Next, let R be the conceptual relation obtained from (I) by replacing each argument that belongs to the set Qb with A1. Let Qr be the set of all arguments in R with the same indexed quantifier as A1. Then, since B1 and A1 satisfy the conditions of Theorem 1, four cases are possible.

(1) A1=B1.

In this case, clearly (I) implies R, since they are identical. Furthermore, the quantifiers of the arguments that belong to the set Qr are all the same, and equal to the indexed quantifier of B1.

(2) B1 is a proper subclass of A1.

To show that (I) implies R, let us assume that relation (I) holds true, that is, that it exists in LTM. If (I) holds true and B1 is a proper subclass of A1, then R is true for some entities in the class A1, namely those in B1. Thus, (I) implies R, and all the arguments in Qr have the same indexed existential quantifier.

(3) A1 is a proper subclass of B1, and A1 inherits the relation R from B1.

If (I) holds true, because of inheritance (I) must hold true for all entities in the class B1 and, in particular, for all entities in class A1, since A1 is a proper subclass of B1. Thus, (I) implies R, and all arguments of R that belong to the set Qr have the same universal quantifier.

(4) C1 is a concept which is a proper subclass of both A1 and B1, and C1 inherits from B1 the conceptual relation S obtained by replacing in (I) each argument that belongs to the set Qb with C1. We let Qs be the set of all arguments in S with the same indexed quantifier as C1.

Applying case (3), above, to the conceptual relations (I) and S, we conclude that (I) implies S, and all arguments of S that belong to the set Qs have the same indexed universal quantifier. Applying case (2) to the conceptual relations S and R, we conclude that S implies R, and all arguments of R that belong to the set Qr have the same indexed existential quantifier. Thus, (I) implies R, where all the arguments of R that belong to the set Qr have the same indexed existential quantifier.

This shows that under the hypotheses of the theorem, conceptual relation (I) implies conceptual relation R. All that remains now is to note that R and (II) may only differ in arguments whose quantifiers are not indexed, so Theorem 2 can be applied to conclude that R implies (II), and therefore that (I) implies (II).

Examples

Let us suppose that the following *action-structure* exists in LTM, and consider the two examples below.

```
(a1
  (args (human)(human))
  (pr (like))
  (actor (human (q1 (allx))))
  (object (human (q2 (allx)))))
```

This structure represents the fact that "everyone likes himself/herself." In the notation of Theorem 3, B1=human, r=like, B2=human, B1 has an indexed universal quantifier, and Qb = { B1, B2 }.

a) Suppose the question "does everyone like everyone?" is asked. This question is represented by the structure

```
(a2
  (args (human)(human))
  (pr (like))
  (actor (human (q (all))))
  (object (human (q (all)))))
```

and in the notation of Theorem 3, A1=human, r=like, A2=human, and the quantifiers of A1 and A2 are universal, but not indexed. Therefore, the relations "A1 r A2" and "B1 r B2" do not satisfy con-

dition (a) of Theorem 3, and the question can not be answered affirmatively.

b) Apart from structure a1, above, let us suppose that the relation underlying the sentence "all americans are humans" also exists in LTM. Consider the question "do all americans like themselves?" This question is represented by the structure

```
(a3
  (args (american)(american))
  (pr (like))
  (actor (american (q (allx))))
  (object (american (q (allx)))))
```

In the notation of Theorem 3, A1=american, r=like, A2=american, A1 has an indexed quantifier, and Qa = { A1, A2 }. Thus, condition (a) of Theorem 3 is satisfied. Since "american" is a proper subclass of "human," and it inherits the relation underlying "all americans like themselves" from "human," condition (b) of the theorem is also satisfied, and the answer to the question is affirmative.

Table 2 contains the algorithm for answering verification questions using the inference criteria we have explained in this section. Let us illustrate the main ideas in the algorithm with an example. Suppose the question asked is "do germs cause diseases?" First, we look in the *object-structure* of *germ*, and ask the question "what do germs cause?" If *disease* is one of the things caused by germs, then the answer will be *yes*. If not, let *c* be something caused by *germs* (as indicated in the *object-structure* of *germ*). If *c* is a *disease* then the answer to the question is *yes*. If not, we determine if *disease* is a descendant of *c*, in which case, if inheritance can be verified, then the answer will be *yes*. If this also fails, we determine if there is a common descendant of *disease* and *c* which inherits the desired conceptual relation from *c*, in which case the answer is *yes*. If all these fail, we examine the ancestors of *germ*. We ask, "is there an ancestor of the concept *germ* which causes diseases and such that *germ* inherits this conceptual relation from it?" If so, the answer will be *yes*. If this also fails, we ask "is there anything which is a germ and which causes diseases?" If so, again the answer will be *yes*. Finally, if this also fails, we try to find out if something which is a *germ* inherits the conceptual relation "cause diseases" from an ancestor other than *germs*.

Consider the verification question represented by the the conceptual relation A given by "A1 r A2,...,Ak." For any conceptual relation, say S, let Qs be either the empty set, if the quantifier of the first argument of S is not indexed, or the set of arguments in S with the same indexed quantifier as the first argument of S.

Initialization Set the list VISITED to nil.

Step 1 Let W be the set of all conceptual relations of the form "A1 r B2,...,Bm" that are present in the *object-structure* of A1 in LTM.
For each conceptual relation w in W, for which Qw = Qa
    For each j in {2,...,k}
        If Aj and Bj are not conceptual relations, then
            If Aj = Bj, or Bj is a subclass of Aj, or Aj is a subclass of Bj and Aj inherits the conceptual relation "A1 r B2,...,Bj,...,Bk" from Bj, or the children of Bj inherit the conceptual relation "A1 r B2,...,Bj,...,Bk" from Bj, and Aj and Bj have a common descendant, then continue with the next j in {2,...,k};
        else if Aj and Bj are conceptual relations and Bj implies Aj, then
            continue with the next j in {2,...,k};
        else continue with the next w in W;
    Answer "yes"; exit;
Add A1 to VISITED.

Step 2 Let PARENTS be the set of concepts *p* such that the conceptual relation "A1 *is-a* p" is present in LTM.
For each *p* in PARENTS
    If *p* is the root of the hierarchy or a member of VISITED, continue with the next *p* in PARENTS.
    Let W be the set of all conceptual relations of the form "p r B2,...,Bm" that are present in the *object-structure* of *p* in LTM.
    For each w in W for which A1 inherits the conceptual relation "A1 r B2,...,Bm" from *p*, and Qw = Qa,
        For each j in {2,...,k}
            If Aj and Bj are not conceptual relations, then
                If Aj = Bj, or Bj is a subclass of Aj, or Aj is a subclass of Bj and Aj inherits the conceptual relation "p r B2,...,Bj,...,Bk" from Bj, or the children of Bj inherit the conceptual relation "p r B2,...,Bj,...,Bk" from Bj, and Aj and Bj have a common descendant, then continue with the next j in {2,...,k};
            else if Aj and Bj are conceptual relations and Bj implies Aj, then
                continue with the next j in {2,...,k};
            else continue with the next w in W;
        Answer "yes"; exit;
    Add *p* to VISITED;
    Append the parents of *p* to PARENTS;

Step 3 Let CHILDREN be the set of concepts *c* such that the conceptual relation "A1 classes-of c" is present in LTM.
For each *c* in CHILDREN
    Perform Step 1 with *c* in place of *A1;*
    Perform Step 2 with *c* in place of *A1;*
    Append the children of *c* to CHILDREN;

Step 4 Answer "I don't know"; exit.

Table 2. Algorithm for Inferring the Answer from the Classification Links in LTM

6. Inferring the Answer by Classifying the Concepts in the Question

In this section, we discuss the inferences drawn when a question is asked about a concept not existing in LTM. Suppose that a question is asked about a concept, say C, that is not present in LTM. Normally, nothing can be said about such a concept. However, if the question itself contains a description of the concept, then such a description may be used by the *Classifier* to determine which concepts in LTM would be the parents and which would be the children of concept C, if C were present in LTM. Knowing what the parents and children of a concept are allows the kinds of inferences discussed in the previous section, even if the concept itself does not have an entry in LTM. We illustrate this situation with an example. Let us suppose that LTM contains the representation of the concepts and conceptual relations underlying the sentences below. This representation is shown in Figure 7.a.

All birds are animals. All animals which live in the Antarctic swim.

Suppose now we ask the question "Do birds which live in the Antarctic swim?" The question concerns the concept *birds which live in the Antarctic*, which does not exist in LTM. However, because the concept is described by the question in terms of other concepts which do exist in LTM (*birds* and *Antarctic*), an attempt can be made to classify the concept in the question with respect to the concepts in LTM. The *Classifier* and *Integration* algorithms are discussed in detail in [10]. We now describe briefly the aspects of the *Classifier* that are relevant to the inference

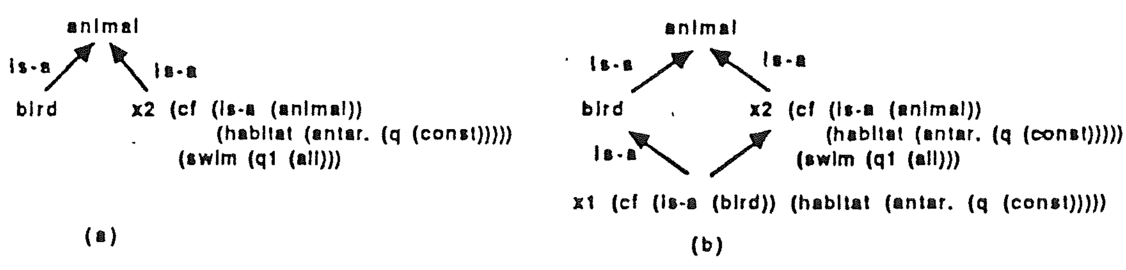


Figure 7. Representation of "All birds are animals. All animals which live in the Antarctic swim." (a) before asking the question "Do birds which live in the Antarctic swim?" (b) after asking the question

C-4

mechanisms.

As shown in Section 2, the concept *birds which live in the Antarctic* is represented by an *object-structure* that has a *characteristic-features* slot, as follows:

(x1 (cf (is-a (bird)) (habitat (Antarctic (q (constant))))))

In order to determine what the parents and children of a concept are, the *Classifier* must be able to compare two concepts and determine whether they are the same, whether one is a subclass of the other, or whether there is no hierarchical relation between them. This comparison is based on the following observations: a) if *x* is a concept whose representation structure contains a *characteristic-features* slot, then a concept *y* represents a subclass of concept *x* if each characteristic feature of *x* is true of all entities in the class corresponding to concept *y*; b) if *x* is a concept whose representation structure does not contain a *characteristic-features* slot, then a concept *y* represents a subclass of concept *x* if the conceptual relation "*y is-a x*" holds true, or if there exists a sequence of concepts *y1,y2,...yn*, such that "*y is-a y1*," "*y1 is-a y2*,"..., "*y(n-1) is-a yn*," and "*yn is-a x*" all hold true.

Given a concept like *x1*, above, whose location in LTM is to be determined, the *Classifier* applies the above criteria to find the parents and children of *x1*. However, the *Classifier* must do this without having to compare *x1* with all or most concepts in LTM. The search for the parents and children of *x1* is first restricted by the contents of the *is-a* slot in the representation structure of *x1*. Since this *is-a* slot contains the concept *bird*, then at least one parent of *x1* must be either the concept *bird* or a descendant of *bird*, and every child of *x1* must be a descendant of *bird*. Thus, the *Classifier* first compares concept *x1* with a relatively small part of LTM, namely, the sub-hierarchy consisting of concept *bird* and its descendants. It is possible though, as our example illustrates, that concept *x1* be related to other concepts in LTM which are not part of the *bird* sub-hierarchy. In general, this might be the largest part of LTM so it is essential to have a way of reducing this search to a minimum number of concepts in LTM. To this objective, we observe the following: a) since all children of *x1* must also be descendants of *bird*, that is, they must be part of the *bird* sub-hierarchy, outside the *bird* sub-hierarchy we can only find parents of *x1*; b) from the representation of *x1* we see that if a concept *y* is a parent of *x1*, then *y* must satisfy the conceptual relation

"*y* habitat *Antarctic*." Therefore, the *Classifier* only needs to examine those concepts in LTM that satisfy this conceptual relation. How can these concepts be found? As we recall from Section 2, every conceptual relation is linked to all of its arguments. Therefore, if the conceptual relation "*y* habitat Antarctic" holds true for some concept *y*, then the *object-structure* of *Antarctic* in LTM must contain the conceptual relation "*Antarctic* habitat-of *y*," and finding the concepts outside the *bird* sub-hierarchy that might be related to *x1* reduces to examining the *object-structure* of *Antarctic*. When this fails, the ancestor concepts of *Antarctic* are also examined. In our example, the *Classifier* determines that concept *x1* is a child of *bird* and also a child of *x2 (animals which live in the Antarctic)*, as shown in Figure 7.b. With this knowledge, it can now be inferred that "birds which live in the Antarctic swim," because "birds which live in the Antarctic are animals which live in the Antarctic" and "all animals which live in the Antarctic swim."

As a second example, suppose the concepts and conceptual relations underlying the following sentences have been stored in LTM:

All mammals are animals. All whales are mammals. All whales live in the sea. Some whales feed on fish.

The representation of these concepts is shown in Figure 8.a Suppose we now ask the question "do animals which live in the sea feed on fish?" Because the question concerns the concept *animals which live in the sea,* which does not exist in LTM, the *Classifier* is activated, which determines that concept *animals which live in the sea* is a child of *animal* and a parent of *whale*, as illustrated in Figure 8.b. With this knowledge, the question can now be answered as follows: "Yes, some animals which live in the sea feed on fish because whales are animals which live in the sea and some whales feed on fish." Similarly, if the conceptual relations:

All infections are diseases. The human-body recognizes germs which cause infections.

are present in LTM then the question "Does the human-body recognize germs which cause diseases?" can be answered by invoking the *Classifier* (which will determine that the concept *germs which cause diseases* is a superclass of *germs which cause infections*), through the following reasoning: "Yes, the human-body recognizes some germs which cause diseases because the human-body recognizes germs which cause infections, and all infections are diseases." These

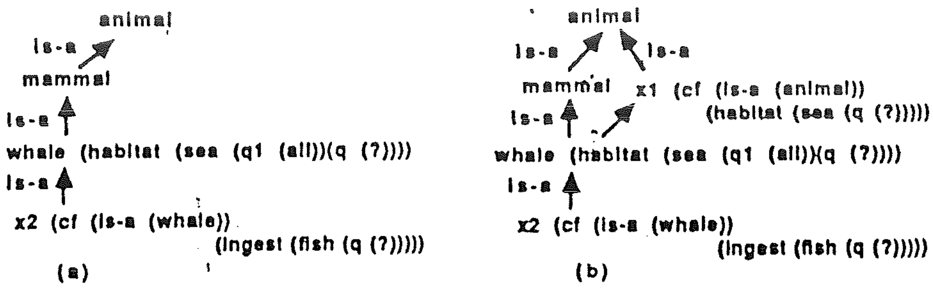


Figure 8. Representation of "All mammals are animals. All whales are mammals. All whales live in the sea. Some whales feed on fish." (a) before asking the question "Do animals which live in the sea feed on fish?" (b) after asking the question

---

examples show how the notion of classification is used as the basis of an inference mechanism when the question involves a concept not present in LTM.

7. Types of Questions Handled by SNOWY

Lehnert [15] identified the following question categories: causal antecedent, goal orientation, causal consequent, verification, instrumental, concept completion, expectational, feature specification, quantification, enablement, disjunction, judgmental and request. Kolodner [14] added time, setting, identification and duration. Dyer [5] included event specification and empathy. We have adopted this categorization with very minor changes. We have added to these categories the following types: definition ("what are bacteria?") and classification ("are bacteria germs?"), a subclass of verification. Also, verification questions of the type exemplified by "do all antibodies kill germs" are further classified as quantificational questions. The present implementation of SNOWY handles the following question categories: verification, classification, definition, feature specification, quantificational, goal, location, instrumental, identification, and concept completion.

The structures below are built for the questions "do all germs cause infections?," "why does the human body produce antibodies?," "does the human body kill germs by producing antibodies?," and "do all americans want to own a house?," respectively.

```
(Verify (germ cause infection
                (a1 (args (germ)(infection))(pr (cause))
                  (actor (germ (q (all))))(object (infection (q (?)))))))

(Purpose (human-body make antibody
                (a1 (args (human-body)(antibody))(pr (make))
                  (actor (human-body (q (?))))(object (antibody (q (?)))))))

(Verify (human-body cause-to-die germ
                (a1 (args (human-body)(germ))(pr (cause-to-die))
                  (actor (human-body (q (?x))))(object (germ (q (?))))
                  (instrument (e1)))
                (e1 (args (human-body)(antibody))(pr (make))
                  (actor (human-body (q (?x)))) (object (antibody (q (?)))))))

(Verify (american want e1
                (a1 (args (american)(e1)) (pr (want))
                  (actor (american (q (allx)))) (object (e1 (q (some)))))
                (e1 (args (american)(house)) (pr (own))
                  (actor (american (q (allx)))) (object (house (q (some)))))))
```

Verification questions fall under the schema *Verify A1 r1 A2*, where A1 and A2 stand for concepts and r1 for relation. Answers for these questions are obtained by searching for the relation *r1 A2* under the concept A1. If a quantifier or a case is part of the question (like in the examples above), those slots in the *action-structure* are examined. For instance, a quantificational question will be answered by searching the *object-structure* for that concept and then by examining the content of the quantifier slots in the *action-structure.* Purpose questions fall under the schema *Purpose A1 r1 A2.* These questions are answered by accessing the "purpose" slot in the *action-structure* representation of the conceptual relation *A1 r1 A2.*

Since conceptual relations are linked to all its arguments, answering identification questions becomes a very straightforward procedure. Thus, for a question like "what causes diseases?" we note that if the conceptual relation "X causes diseases" is true for some concept "X," then each such relation will also be indexed under the concept "diseases," so when trying to answer this question, we look in the LTM entry for "diseases" under the relation "cause:by." Anything found in this slot will be an answer to the question. Likewise, consider the sentence "the human body destroys germs by producing antibodies." The conceptual relation described by this sentence is linked to "human body," to "germs," and to "antibodies," so it can be accessed through any of these concepts. If this were not so, questions like "what do you know about antibodies?" or "for what purpose does the human body produce antibodies?" could not be answered.

8. Discussion

In summary, we have shown a method to represent and quantify in relations with multiple arguments, some of which can also be relations. We have also indicated how necessary, necessary and sufficient, and contigent information can be associated with a given concept. We have provided FOPC formulae for each of the knowledge structures presented in this paper. Finally, we have proven that the algorithm draws valid inferences from the representation structures.

The representation language which has been described here is a subset of FOPC. However, the interesting aspect of this language lies in the reasoning algorithms which it permits and in its representation capabilities. It is also relevant to note that one of the main motivations behind the design of this language has been the acquisition of knowledge from texts, which involves the construction of knowledge representation structures from scratch. And this in turn requires recognition and integration algorithms which will be very hard to accommodate into a knowledge representation scheme based exclusively on first order logic.

One intriguing aspect of these ideas is that deep semantic relations such as classification hierarchies underly syntactical constructions like restrictive relative clauses. The relevance of explicit classification has been widely observed by cognitive scientists working on comprehension. In explicit classification, the classes are clearly introduced and named, as in the passage:

> **There are two kinds of mammals: terrestrial mammals and aquatic mammals. Aquatic mammals include the carnivorous and the omnivorous.**

The classification underlying restrictive relative clauses is a more basic cognitive phenomenon than explicit classification. In this paper, we have shown that it is a necessary element in deriving certain kinds of inferences. In [11], we have shown that this type of classification is also an essential component for the acquisition of knowledge from texts.

One may wonder if concepts underlying relative clauses and other restrictive qualifiers should have the same status in memory as concepts for which "we have words." We assume that "we have words" means single words, since we have indicated that complex noun groups are also represented as classification hierarchies. The answer is an unqualified "yes." The concept *car* is expressed currently by a single word, namely "car." The concepts expressed by the phrases "cars

with carburetors" and "cars with fuel injection" are as natural as the concept *car*. If these concepts are not classified under the concept *car*, many inferences could not be accomplished and, as a consequence, understanding will not take place.

From a methodological point of view, this paper is not a logicist paper, since the program which embodies the theory does not use logic to derive inferences. In fact, a main emphasis in this research has been to discover alternative methods to logic, which integrate indexing and inference mechanisms. We have shown that certain deductive problems can be solved by the right representation of knowledge and by keeping memory organized in a principled manner. The boundaries of logic are not the boundaries of rationality. However, we do think that logic is an excellent tool to convey the meaning and scope of theories.

One of the main applications of these ideas, which also constitutes a natural theoretical extension, is to the task of knowledge acquisition from texts for expert systems. We are presently building a knowledge acquisition system which builds automatically from texts certain kinds of expert systems. The domain expert is instructed to describe its subject in a top down manner. The system builds automatically the classification hierarchies from the text and, then, the reasoning techniques explained in this paper are used to answer questions posed by users of the expert system. In this system, comprehension, memory reorganization and problem-solving are integrated processes, since all of them share the same algorithms. (See [11] for a discussion of these ideas.)

Acknowledgement

This paper has benefited from the insightful and critical comments of an anonymous referee. He/she is gratefully acknowledged.

References

[1] J. Allen and C. R. Perrault, "Analyzing Intention in Utterances," *Artificial Intelligence*, vol. 15 (1), pp. 143-178, 1980.

[2] R. Brachman and J. Schmolze, "An Overview of the KL-ONE Knowledge Representation System," *Cognitive Science*, vol. 9, pp. 171-216, 1985.

[3] R. Carnap. *Meaning and Necessity*. University of Chicago Press, 1947.

[4] E. Charniak and D. McDermott, *Introduction to Artificial Intelligence*. Reading: Addison-Wesley, 1985.

[5] M. Dyer, *In-Depth Understanding*. Cambridge: MIT Press, 1983.

[6] F. Gomez, "Towards a Theory of Comprehension of Declarative Texts," in *Proceedings of the 20th Meeting of the Association of Computational Linguistics*, Toronto: ACL, 1982, pp. 36-43.

[7] F. Gomez, "A Model of Comprehension of Elementary Scientific Texts," in *Proceedings of the Workshop on Theoretical Approaches to Natural Language Understanding*, Halifax, Nova Scotia, 1985, pp. 70-81.

[8] F. Gomez, "WUP: A Parser Based in Word Usage," in *Proceedings Phoenix Conference on Computers and Communication*, Scottsdale: IEEE, 1988, pp. 445-449.

[9] F. Gomez and C. Segami, "SNOWY: A System Which Understands Elementary Scientific Texts," Technical Report CS-TR-87-04, Department of Computer Science, University of Central Florida, Orlando, Florida, 1987.

[10] F. Gomez and C. Segami, "The Recognition and Classification of Concepts in Understanding Scientific Texts," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 1, pp. 51-77, 1989.

[11] F. Gomez, "Knowledge Acquisition From Natural Language For Expert Systems Based On Classification Problem-Solving Methods," Fourth AAAI-Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop," Banff, Canada, 1989.

[12] B. J. Grosz, A.K. Joshi, and S. Weinstein, "Providing an Unified Account of Definite Noun Phrases in Discourse," In Proceedings, ACL, 1983, pp. 44-50.

[13] B. Grosz and C. Sidner, "Attention, Intention, and the Structure of Discourse," *Computational Linguistics,* Vol. 12, 1986.

[14] J. Kolodner, *"Retrieval and Organizational Strategies in Conceptual Memory,"* Technical Report 187. Dept. of Computer Science, Yale University, 1980.

[15] W. Lehnert, *The Process of Question Answering.* Hillsdale: Lawrence Erlbaum, 1978.

[16] Litman, D.J. and J.F. Allen. "A Plan Recognition Model for Subdialogues in Conversations," *Cognitive Science,* vol. 11,2, pp. 163-200, 1987.

[17] W. V. O. Quine, "On what there is," in *From a logical point of view.* Cambridge: Harvard University Press, 1964.

[18] R. Schank, *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, 1982.

[19] J. Schmolze and T. Lipkis, "Classification in the KL-ONE Knowledge Representation System," in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence,* Karlsruhe: IJCAI, 1983, pp. 330-332.

[20] B. L. Webber, "The Interpretation of Tense in Discourse," in Proceedings, ACL, 1987, pp. 147-154.

[21] R. Wilensky, *Planning and Understanding.* Reading: Addison-Wesley, 1983.

S5-54
333525
N91-70703

UCF / KSC COOPERATIVE AGREEMENT

PROJECT #6

PRODUCTIVITY TECHNIQUES

YEAR 1989- 1990 REPORT

March 1991

Table of Contents

1. INTRODUCTION

The preliminary study conducted in summer 1989 resulted in the identification of the following productivity techniques which would have a high potential of enhancing the productivity at KSC:

1. The design and prototyping of an ergonomic work stand for overhead operations performed on the orbiter.

2. The use of Voice Data Entry for potential operations at KSC.

3. The study of the use of laser and/ or optical based devices for cavity digitization.

There were a number of other techniques and potential technologies which were identified as potential contributors to the productivity enhancement at KSC. The study for such technologies was deferred to the next phases of the study. Those areas are:

1. The use of Hypermedia in support of operations at KSC.
2. The study of the Data quality at KSC and the devising of "early indicators" based on statistical analysis for potential problems.
3. The effect of cross training vs. specialization on the productivity and quality of work at KSC.

This document reports on the first years effort of the study. The report builds on the analysis conducted in the summer of 1989 and which was documented in a previous report submitted to KSC at the conclusion of that study.

In this executive summary we summarize the outcome of the overall study. Each of the individual study areas is documented separately in the accompanying reports in the Appendices.

2. WORK STAND DESIGN AND PROTOTYPING

The summer 1989 study resulted into three conceptual designs for a two-person workstand as well as a set of guidelines for the detailed design. Such designs were formulated after studying a number of operations conducted on the orbiter. The main feature in all of the designs pointed to the need for a back support in any design especially for long duration activities which would require positioning of the operator in a non natural posture - overhead operations. The objectives of this year's study is to develop a detailed design for a complete workstand, build a prototype for the workseat, validate and evaluate the prototype through a laboratory experiment. The final report for this phase is in Appendix I.

2.1 The Detailed design:

Two detailed designs were produced. The first one features a hydraulic lift to raise an ergonomic workseat to a working area under the orbiter. The seat can be adjusted to a variety of positions using a controlling joy-stick which controls the back-rest, headrest and the lift. The space around the seat would allow for another worker (helper) to operate. The advantage of this design is that it is flexible enough to accommodate a variety of body sizes and working positions. The main disadvantage of this design is inflexibility with respect to a coverage of larger working area without changing the stand position on the floor. The estimated cost for a prototype based on this design is around $50,000. Subsequent units should be less in price.

The second design features a two seat adjustable workstand without platform. Based on the results of the workseat experiment (see Section 3) using actual tile technicians and typical OPF operations, it was determined that a helper would not be needed if a worker has all the necessary tools, equipment, tool, and material within reach. This eliminated the need for a platform. The design provides for an access to all necessary accessory for an operation as well as protection for any material or chemical dripping on the worker while performing an operation. In a typical operation, the user would sit on the seat and reach for the desired height by the use of the lifting controls to perform an operation. Raising the seat to the proper height is accomplished through either hydraulic or electrical lift.

2.2 The Ergonomics Seat:

Based on an initial study a workseat was fabricated with the intention of using it on an existing workstand in an ergonomic experiment. The seat is made up of steel pipes without any controls or adjustment capabilities. The seat fabrication was done through subcontracting with a local steel manufacturing company. The seat was mounted on an existing two-person workstand,obtained from NASA-KSC. The experiment was setup at

the Space Technology Institute (STI) of Brevard Community College (BCC) in Cocoa. STI conducts classes for tile operations as part of training program for tile technicians. The laboratory setup was coordinated with NASA-KSC project officer.

2.3 Experimental Analysis:

An experiment was-conducted using the fabricated ergonomics seat and the workstand setup at the Space Technology Laboratory of BCC. The objective of this experiment was to validate the seat dimensions and its effectiveness in improving worker performance and reducing body fatigue. Overhead working environment was created by suspending work surface from the ceiling at the height of about 10 feet from floor which simulates orbiter operations. A number of technicians were recruited from KSC, necessary material, equipment and tools were secured from both IST and KSC. The details of the experiment including design, conduct, findings and results are documented in the accompanying report (Appendix I).

The results of the experimental evaluation revealed that ergonomically designed workseat improves the work performance and provide body comfort during overhead operations especially for long duration activities. More specifically, the following summarize the evaluation results.

1. The seat dimensions were suitable for the conduct of the majority of operations except the lack of head and neck mobility offered by the workseat. However, the original workseat design include the adjustable headrest which was not implemented in the prototype due to cost constraints.

2. The worker's endurance time was significantly increased when the workseat was used. There was an average of 63 percent increase in endurance time.

3. Task accuracy was significantly increased when the workseat was used. This reflects directly on the quality of work.

4. There was no significant difference, with and without workseat in the task completion time. This may be due to relatively short time duration required to complete the experimental tasks. However, when coupled with the large differences in endurance time, there will be a significant difference in task completion times over the course of long work hours.

5. The workseat greatly reduced worker's body discomfort caused by overhead positions. In some specific body parts such as lower legs and thigh, the body discomfort was eliminated.

Based on the above mentioned findings, it is concluded that the ergonomically designed workseat is effective in improving the productivity of tile technicians and in reducing their body fatigue.

The tile technicians' general comments were also compiled during the use of workseat. The following are some comments, which would affect the prototype buildup, shared by most of tile technicians during the use of workseat.

(1) There was not enough clean work area, tools and equipment storage, and waste container. A tool "tray" or some other means for the operator to have access to the tools without changing position is necessary.

(2) The technicians concerned over dripping RTV and other chemicals as well as falling debris. A protection such as a face shield is desirable.

(3) There is a need for information access while work is being done for some operations as well as exchanging information with floor crew.

(4) There is a need for an arm rest in each side.

3. VOICE DATA ENTRY (VDE)

The previous effort in the summer 1989 resulted in the development of a training and implementation strategies for the VDE in support of the "step and gap" operation. Such application for the VDE was identified and the software for it was developed by LSOC and Stanford university as part of the SIORA program. Since that time the interest of NASA-KSC and the UCF project team has shifted to different objectives. Specifically, the objectives of this phase, as outlined in working paper appendix II.1, are to study the feasibility of incorporating a microcomputer based VDE with the SPDMS II at KSC and to identify new applications at KSC which may benefit from the VDE technology. While the details of the study are documented in the attached report in Appendix II.2. The following is a summary of the outcome of the study with respect to achieving the two objectives stated above:

3.1 The Feasibility of a Microcomputer based VDE system at KSC:

The study in this area was in two fold. The first involved the acquisition of a typical microcomputer based VDE and conducting a typical application experiment at UCF. The second involved a study of the proposed SPDMS II and identifying and resolving any potential compatibility problems. The outcome of the feasibility study indicates that the VDE technology is not reliable enough for the moment to contribute to an increase in the efficiency or the productivity of any of the operations at KSC. However once the technology improves it may have a positive impact on the productivity of a variety of operations at KSC which involve man-computer interaction. This outcome was demonstrated through a laboratory experiment conducted at UCF, the details of which are documented in the accompanying report.

It was also determined that there is no effect on the computer type (Micro vs. otherwise) on the performance of the VDE system. In fact microcomputers are probably better to use for VDE applications since the majority of voice recognition can be done on the micro level before interacting with larger computers for the purpose of data entry or information retrieval.

The shortcomings in the technology cited in the report is in continuous speech recognition, the limitation of the acceptable vocabulary, user dependency, training time, and the remote communication between the user and VDE system which is not always a possible requirement in the KSC environment. Such limitations have to be overcome before such technology would be feasible for use in KSC operations in a manner that will increase productivity and enhance performance.

Studying the proposed SPDMS II and a similar system at UCF reveals no problems of interfacing SPDMS II with any VDE. Most of the commercially available VDE can easily

be interfaced with SPDMS II. In rare cases a software interfacing program may have to be acquired or developed to facilitate the dialogue between the system elements.

3.2 Potential Application at KSC:

A model was developed to aid in the identification and evaluation of potential jobs which may be amiable to the use of VDE technology. The model considers the technological demands for a successful job performance, as well as the benefits which may materialize if VDE is used. Factors such as the working conditions, need for free hand environment, job duration, the vocabulary used in performing the operations, training requirement and the impact on the productivity are incorporated in the model. Jobs are evaluated against the various factors based of a point scale and the viability of the VDE application is assessed. The model has been used in identifying "Problem Reporting" as one such operation which may benefit from the VDE technology. A parallel effort is being conducted in identifying other jobs and in collecting the necessary vocabulary for the Problem Reporting prototyping.

4. CAVITY DIGITIZATION

The summer 1990 effort resulted in demonstrating the technical feasibility of obtaining a 3-D mapping of a cavity through the use of optical/ laser based technology. The study also identified equipment manufacturers and potential problems of implementation in the KSC environment. A grant from FHTIC to UCF provides partial support for this component of the study and calls for UCF to team up with an industry for system development.

The objective of the next phase of the study is to build and demonstrate a device which through the use of laser and/or optical technology would collect 3-D descriptive information of a tile cavity in the orbiter and display it in a form that would enable the fabrication of a replacement tile. However the effort is continuing, the following is a point summary of what have been accomplished so far. The details of the effort is the accompanying report (Appendix III):

1. A joint project between LSOC and UCF was initiated. The project team is composed of 4 UCF researchers and 2 LSOC researchers. The basis for the project was an interim report submitted to NASA-KSC and LSOC. Appendix III.1 is a copy of the interim report.

2. A complete plan for the project, its activities and durations, estimated cost and the responsibilities of each partner was detailed and approved by the project leaders. Appendix III.2 is the details of the plan.

3. UCF pledged a partial fund of $50,000 for the project through a grant from Florida High Technology and Industry Council (FHTIC) and LSOC pledged $125,000 for the project from its own R&D funds.
Appendix III.3 is LSOC commitment.

4. Two types of technologies have been identified as feasible technologies (a) Laser scan based system and (b) Image processing based system.

5. For each of the two systems above, technology developers were identified as the National Research Council of Canada (NRCC) and its licensees for the laser based technology and Lockheed Missile and Space Co. (LMSC) for the image processing based technology.

6. Technical teams have visited the two companies where the technology are being developed. On each visit the technology was demonstrated to the team.

7. Currently the project team is in the process of evaluating the technology and soliciting proposals from the respective companies.

8. In a parallel effort UCF has acquired a laser probe and a controlling mechanism for the purpose of experimenting on the use of the lasers in obtaining 3-D images and for possible use in the quantification of orbiter oscillation. Work is in progress in this area. Appendix III.3 is a tally of the experiments in this area.

5. TECHNOLOGY TRANSFER

A total of 12 technical papers and conference participation was conducted by the research team over the span of the last year. Following is a tally of the paper's titles and conferences. Sample publications are in Appendix IV.

1. Cindy L. Mollakarimi, Tamim S. Hamid, (Lockheed), UCF project recognized, "Remote Voice Training: A Case Study on Space Shuttle Applications," AVOIS '89 Conference Proceedings and presentation, September 12-14, 1989, Los Angeles, CA.

2. Yasser A. Hosni, John Scarboro, (UCF), Tamim Hamid (Lockheed); 12th Conference on Computers and Industrial Engineering, March 12-14, 1990, Orlando, Florida.

3. Tim Barth, Yasser Hosni, and William Swart, "Improving Productivity of Space Shuttle Processing," Proceeding IIE Aerospace and Defense Division 13th Annual Spring Conference, Orlando, FL, February 14, 15, 16, 1990.

4. Yasser A. Hosni, Chin H. Lee, (UCF), Timothy S. Barth, and Cedric Hill, (NASA-KSC), "Ergonomically Designed Workstand for Overhead Operations: Case of Heat-Tile Replacement in the Space Shuttle," Human Aspects of Advanced Manufacturing and Hybrid Automation, 2nd International Conference, Honolulu, Hawaii, August 12-16, 1990.

5. Yasser Hosni (UCF), Tamim Hamid (Lockheed); "Speech Recognition and Synthesis, Training and Implementation Strategies"; Human Aspects of advanced Manufacturing and Hybrid Automation, 2nd International conference; Honolulu, Hawaii, August 12-16, 1990.

6. Yasser A. Hosni, John Creech (UCF), Timothy S. Barth, and Cedric Hill, (NASA-KSC); "Workstand Design for Overhead Operations; Case of Heat-Tile Replacement in Space Shuttle"; 1990 International Industrial Engineering Conference; San Francisco, CA, May 20-23, 1990.

7. Presentation: "Ergonomic Workstation for Overhead Operations" Innovation 91. The 4th Annual Conference of Universities, Industry, Entrepreneurs & Government for Joint Technology Commercialization. Evaluate for innovation 90, however will be considered for Innovation 91 upon completion of prototype.

8. Yasser A. Hosni, Daniel Nasser, Jueng-Shing Hwang, and Labiche Ferreira; "Non-Contact, 3-D, Object Digitizing Systems for Die-Manufacturing"; Paper submitted for the 1990 Society for Integrated Manufacturing Conference, San Antonio, Texas, October 28-31, 1990.

9. Yasser Hosni, Jueng Shing Hwang and Labiche Ferreira, "Tool Path Generation From Surface Mapping of an Object", PROCIEM '90, Tampa, Florida, November 14-16 1990.

10. Yasser Hosni, Daniel Nasser, Jueng-Shing Hwang and Labiche Ferreira, "Non-Contact, 3-D, Object Digitizing Systems for Die Manufacturing", Computers and Industrial Engineering Conference, Orlando, Florida, March 11-13 1991.

11. Yasser Hosni, Tamim S. Hamid and Andrew E. Okraski, "Hypermedia Based System for Space Shuttle Processing", Computers and Industrial Engineering Conference, Orlando, Florida, March 11-13 1991.

12. Chin H. Lee, Yasser Hosni, Lisa Guthrie (UCF), Timothy Barth, and Cedric Hill (NASA-KSC), "Design and Evaluation of a work seat for Overhead Operations", International Industrial Ergonomics and Safety Conference '91, Lake Tahoe, NV June 10-14, 1991.

6. PROJECT TEAM

Over the span of the last year a total of four faculty members and 8 graduate students have participated in the project activities. Following is a tally of the project team, responsibilities, and status of each:

1. Dr. Yasser Hosni, Professor, P.I. and Project Director

2. Dr. William Swart, Professor, Co-P.I.

3. Dr. Chin Lee, Asst. Professor, Workstand Component

4. Dr. Robert Safford, Visiting Associate Professor, Voice Data Entry and Cavity Digitization.

5. Mr. Jueng Shing Hwang, Cavity Digitization, M.S.I.E. (Computer Integrated Manufacturing), Expected graduation date - Spring 1991.

6. Mr. Labiche Ferreira, Cavity Digitization, M.S.I.E. (Computer Integrated Manufacturing), Expected graduation date - Spring 1991.

7. Mr. Thomas Pax, Cavity Digitization, M.S.Computer Engineering, Expected graduation date - Fall 1991.

8. Mr. Yuanlin Shi - Cavity Digitization, M.S.Computer Engineering, Expected graduation date - Fall 1992.

9. Mr. Kenneth Cole - Voice Data Entry, M.S.I.E. (Engineering Administration), Expected graduation date - Fall 1991.

10. Miss Lisa Guthrie - Workstand Design, M.S.I.E. Expected graduation date - Fall 1991.

11. Mr. Andrew Okraski - Hypermedia Applications, Ph.D. Industrial Engineering.

12. Mr. Tamim Hamid - Hypermedia Applications, M.S.I.E. (Computer Integrated Manufacturing), Expected graduation date - Spring 1991.

    In addition, 3 students graduated in the phase of the project: 1989-1990.

    Two researchers form LSOC have joined the team in support of the Cavity Digitization component.

7. APPENDICES

Appendix I. Workstand Design Component

Appendix I.1 Working paper for the workstand design
Appendix I.2 Workseat Experiment Report

Appendix II. Voice Data Entry

Appendix II.1 Working paper for the VDE component
Appendix II.2 Voice Data Entry Technical Report

Appendix III. Cavity Digitization Report

Appendix III.1 Cavity Digitization Interim Report
Appendix III.2 UCF/LOSC Project Plan
Appendix III.3 Tally of experiments for laser equipment at UCF.

Appendix IV. Sample Publications

Appendix I. Workstand Design Component

Appendix I.1 Working paper for the workstand design

Appendix I.2 Workseat Experiment Report

Appendix I.1 Working paper for the workstand design

UNIVERSITY OF CENTRAL FLORIDA
INDUSTRIAL ENGINEERING PROJECT
UCF/KSC COOPERATIVE AGREEMENT

WORKING PAPER FOR PHASE II - FALL 1990 - SPRING 1991

WORKSTAND DESIGN COMPONENT

OBJECTIVE:

This working paper is to report the objective, procedure, and outcome for the phase III of the study. The objectives of the study phase are:

- To design an ergonomic workstand for overhead operations performed on the orbiter.
- To produce a prototype of the workseat which is a main component of the workstand.
- To validate the workseat design and evaluate its effectiveness through a laboratory experiment.

PROCEDURE:

- Set up experimental evaluation at Brevard Community College in Cocoa.
- Recruit tile technicians.
- Conduct experimental evaluation of the prototype of workseat using tile technicians as subjects.
- Analyze the experimental data.
- Document experimental evaluation and data analysis in a technical report.

OUTCOME:

1) Workseat evaluation data.
2) Technical paper documenting the findings of this phase.
3) Improved design.

UCF TEAM INVOLVED: Dr. Yasser Hosni, Dr. Chin Lee, Lisa Guthrie

Appendix I.2 Workseat Experiment Report

UNIVERSITY OF CENTRAL FLORIDA/KSC
COOPERATIVE AGREEMENT

PROJECT #6

PRODUCTIVITY TECHNIQUES

Workstand Design Component

Phase II Report

Spring 1991

Table of Contents

WORKSTAND DESIGN, PROTOTYPING, AND EVALUATION

The summer 1989 study resulted into three conceptual designs for a two-person workstand as well as a set of guidelines for the detailed design. Such designs were formulated after studying a number of operations conducted on the orbiter. The main feature in all of the designs pointed to the need for a back support in any design especially for long duration activities which would require positioning of the operator in a non natural posture - overhead operations. The objectives of this year's study is to develop a detailed design for a complete workstand, build a prototype for the workseat, validate and evaluate the prototype through a laboratory experiment.

1.0 Detailed Design

Two detailed designs were produced. The first one is that shown in Figure 1 (A and B). The main features include a hydraulic lift to raise an ergonomic workseat to a working area under the orbiter. The seat can be adjusted to a variety of positions using a controlling joy-stick which controls the back-rest, headrest and the lift. The space around the seat would allow for another worker (helper) to operate. This design is the result of several levels of improvement above the initial design developed in previous phases. A hydraulic lift to raise the ergonomic workseat to the 8' to 12.5' high work areas under the orbiters is a major design feature. This design takes the individual into account in the initial stages of design by considering anthropometric, biomechanical, physiological, and anatomical properties of the individual to reduce fatigue, improve efficiency, and thus enhance performance.The adjustable workseat can assume a variety of supported positions with a joy-stick which controls the back-rest, headrest, and the lift. The segment of the chair supporting the lower leg is also manually adjustable to accommodate a larger population size range.

The advantage of this design is that it is flexible enough to accommodate a variety of body sizes and working positions. The main disadvantage of this design is inflexibility with respect to a coverage of larger working area without changing the stand position on the floor and high cost. The estimated cost for a prototype based on this design is between $50,000 and $75,000 depending on the type of material used. Subsequent units should be less in price.

The second design is shown in Figure 2. This design features a two seat adjustable workstand without platform. Based on the results of the workseat experiment using actual tile technicians and typical OPF operations, it was determined that a helper would not be needed if a worker has all the necessary tools, equipment, tool, and material within reach. This eliminated the need for a platform. Ergonomically designed workseats will be mounted on a commercially available scissor lift. This design will include all necessary accessories for an operation as well as protection for any material or chemical dripping on the worker while

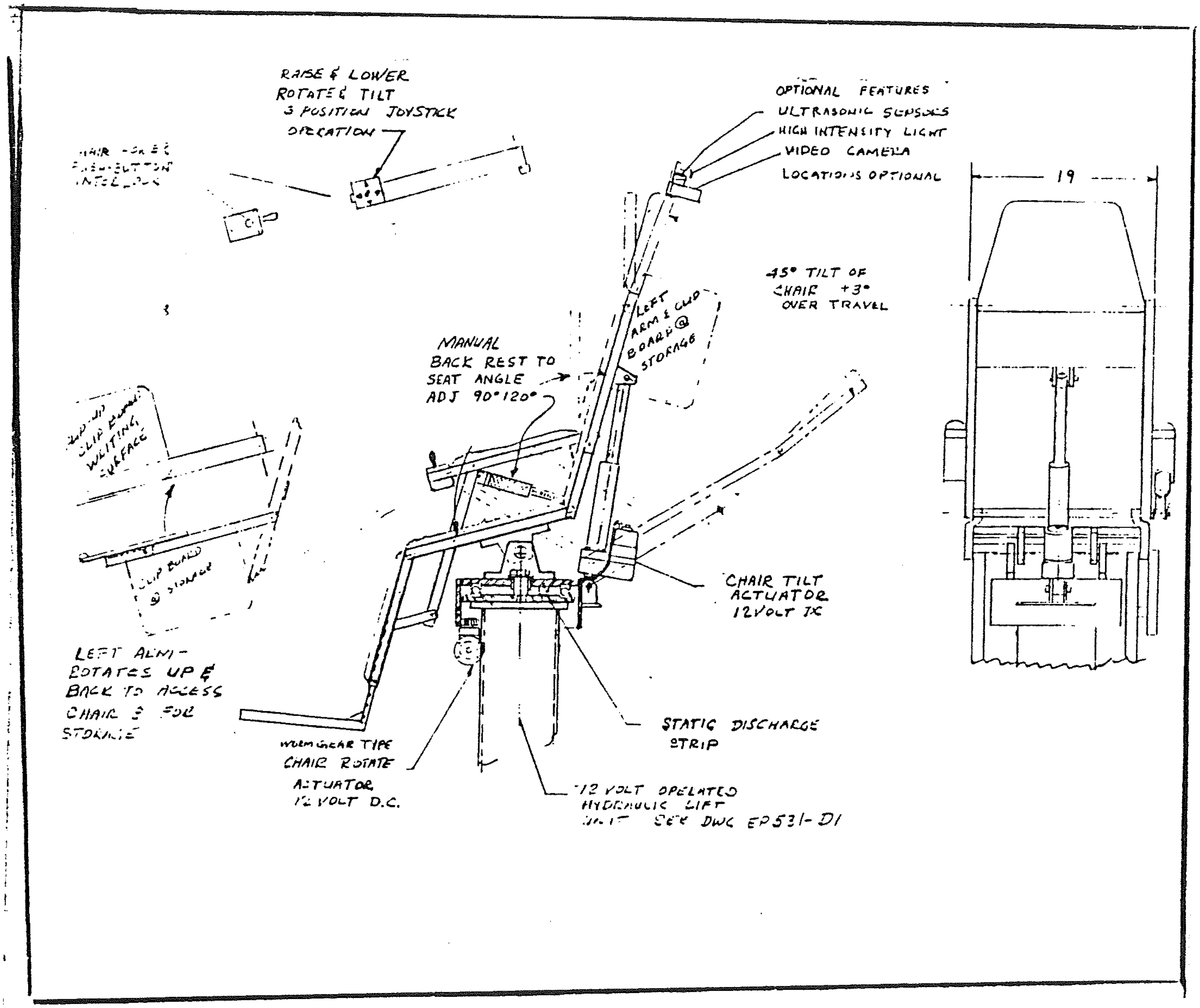


Figure 1A. Workseat; Detailed Design 1

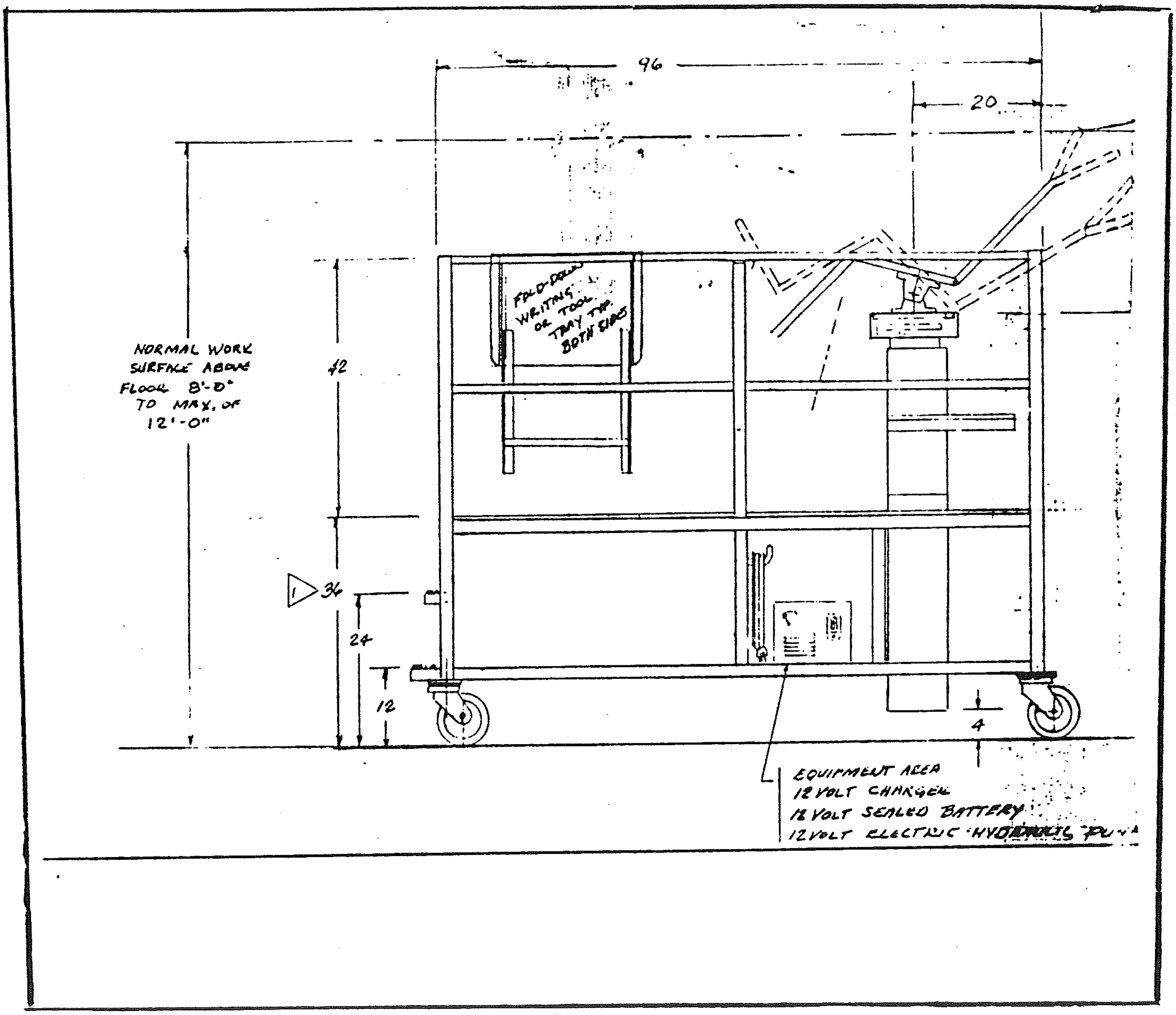


Figure 1B. Workstand; Detailed Design 1

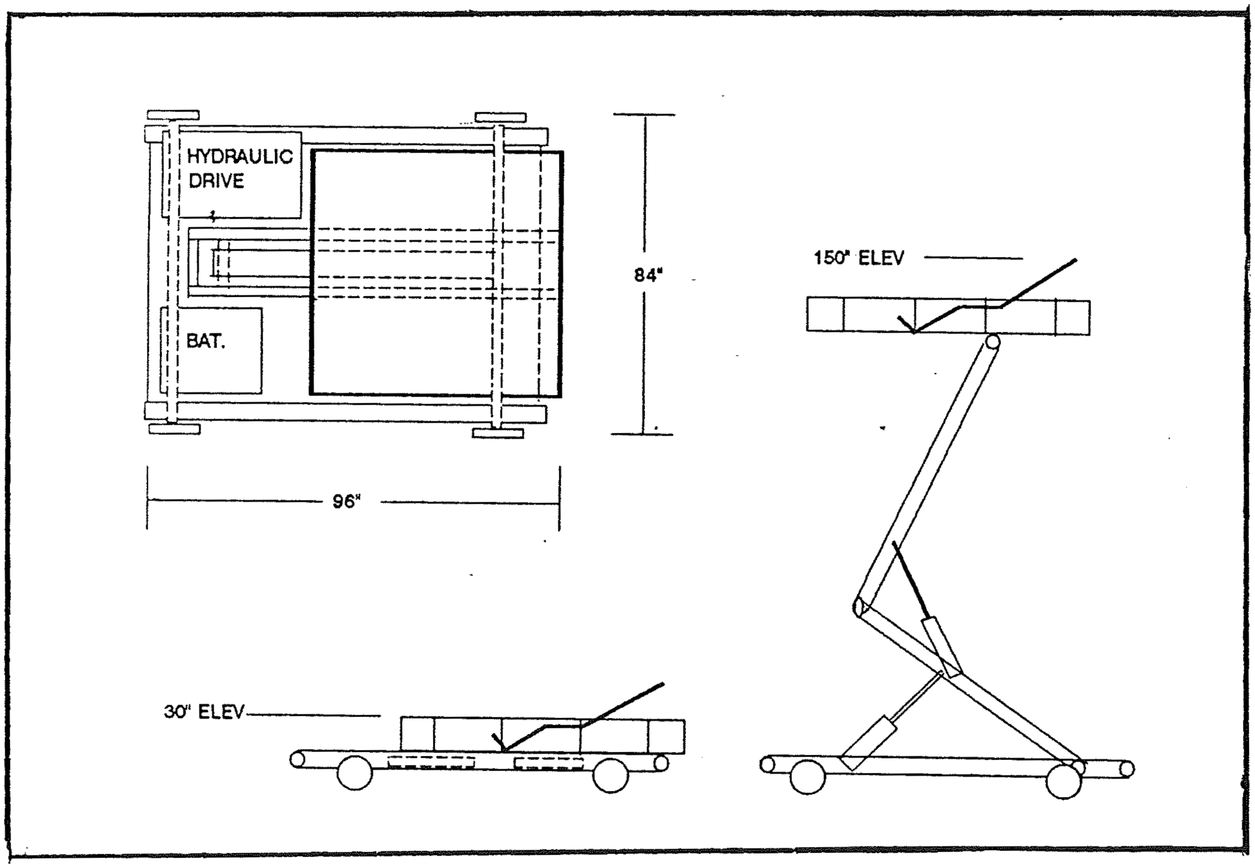


Figure 2. Workstand, Detailed Design 2

performing an operation. In a typical operation, the user would sit on the seat and reach for the desired height by the use of the lifting controls to perform an operation. Raising the seat to the proper height is accomplished through either hydraulic or electrical lift. The cost for a unit is expected to be approximately $25,000.

2.0 The Prototyping of the Ergonomic Workseat:

Based on an initial study a workseat was fabricated with the intention of using it on an existing workstand during the experimental evaluation. The seat is made up of steel pipes without any controls or adjustment capabilities. Figure 3 shows the seat design and its dimensions. The seat fabrication was done through subcontracting with a local steel manufacturing company. The seat was mounted on an existing two-person workstand, obtained from NASA-KSC, as shown in Figure 4, and evaluated through an experiment conducted at the Space Technology Institute (STI) of Brevard Community College (BCC) in Cocoa.

3.0 Experimental Evaluation:

An experiment was conducted using the fabricated ergonomic workseat and the workstand setup at the STI of BCC. STI conducts classes for tile operations as part of training program for tile technicians. The laboratory setup was coordinated with NASA-KSC project officer. The objective of this experiment was to validate the seat dimensions and its effectiveness in improving worker performance and reducing body fatigue. Overhead working environment was created by suspending work surface from the ceiling at the height of about 10 feet from floor which simulates orbiter operations. A number of technicians were recruited from KSC, necessary material, equipment and tools were secured from both STI and KSC.

3.1 Evaluation Techniques

The experimental evaluation consisted of both objective and subjective evaluation; human performance measurement and subjective evaluation. The human performance measurement serves as a method for quantitatively measuring the effects of the implementation of the proposed workseat by providing an accurate way to measure changes in performance parameters. The subjective evaluation reveals very useful qualitative information associated with discomfort and muscle fatigue by measuring the levels of discomfort associated with each posture during the performance of the experimental tasks.

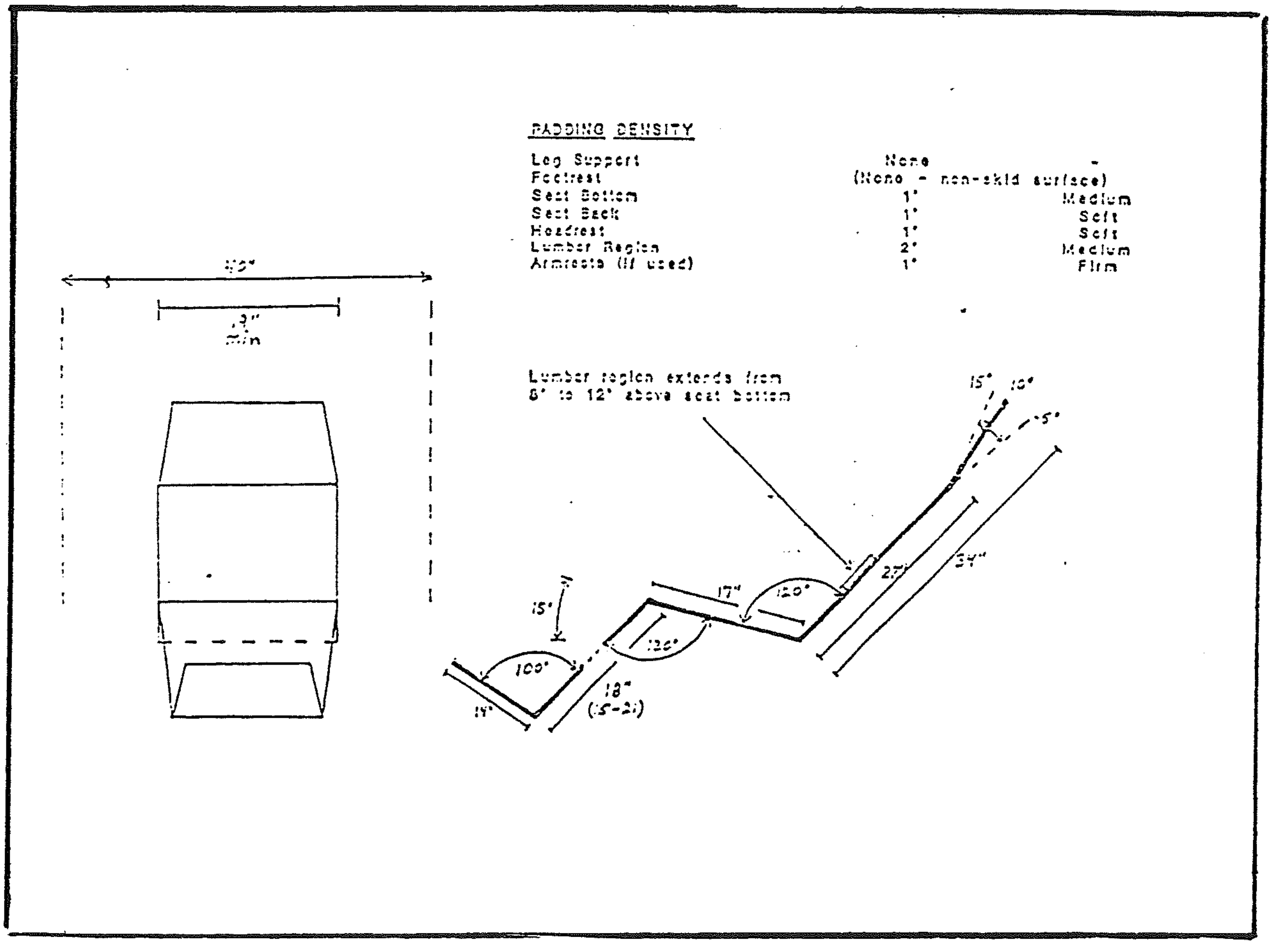


Figure 3. Ergonomic Workseat; Design and Dimensions

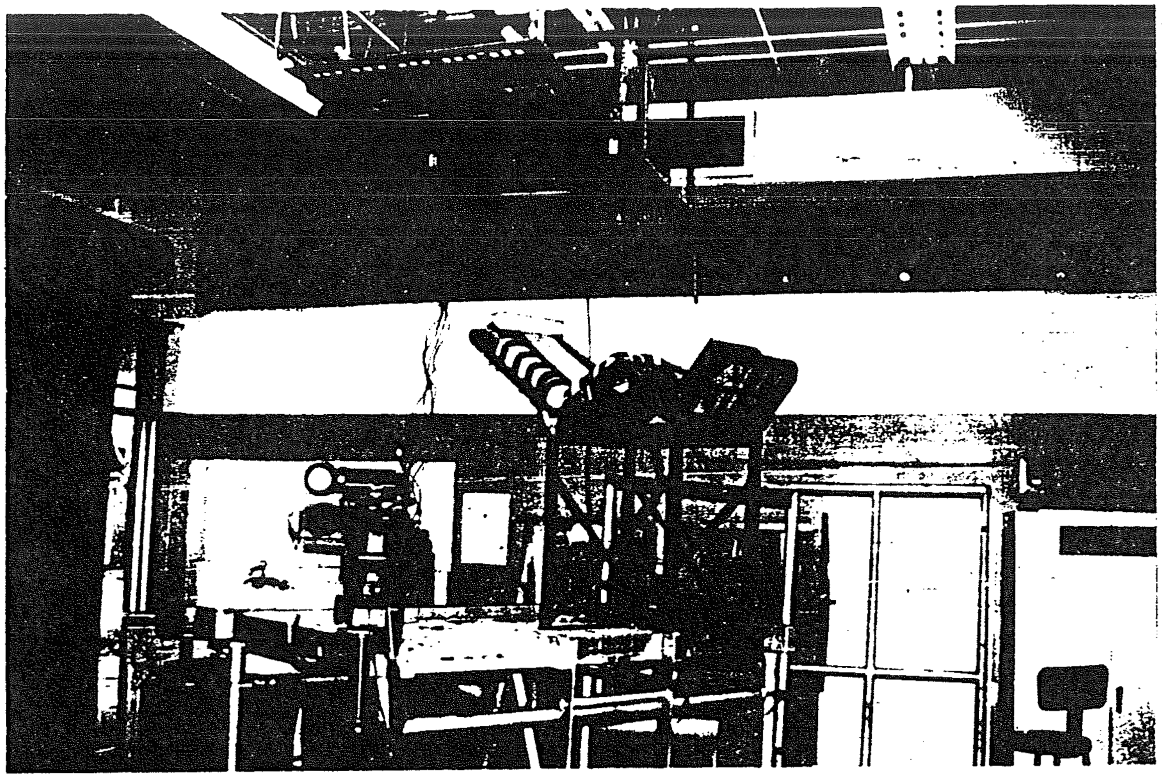


Figure 4. Ergonomic Workseat on Workstand

3.2 Experimental Setup

With the cooperation of the IST at Brevard Community College and the tile technicians recruited from KSC, the experimental evaluation was conducted by simulating the actual Thermal Protection Repair and Maintenance Procedures as well as two additional general application tasks. Thus, the evaluation results will be a direct reflection of the possible improvement of workers at KSC and others performing similar overhead operations elsewhere.

Process P-601 **Installation of Pillow (Alumina) Gap Fillers** and P-301 **Tile Installation** were simulated because of the high frequency of their occurrence at the OPF and ease of simulation. The simulated orbiter surface for Processes P-601 and P-301 is shown in Figure 5. Figures 6 and 7 show a subject performing the processes. In addition, **Rotary Pursuit Tracking** and **Weighted Overhead Work** were also included to simulate general overhead task applications.

Rotary Pursuit Tracking is designed to allow for human performance measurements in the area of accuracy by simulating fine motor skills and is a commonly accepted and recognized method of measurement in the ergonomics field. Simply, the task involves a fluorescent light source rotating in a circular motion and a stylus wand with a photocell at the tip as shown in Figure 8. The subjects pursued the moving light target with the stylus, attempting to keep the stylus and light moving together on the circular path. Figure 9 shows a subject performing the tracking task in the standing position. The photocell initiates the digital stop clock shown in Figure 10. The clock measures the cumulative time of successful pursuit. By keeping the time of the task, sensitivity of the sensor, and speed of rotation constant for all subjects, an effective measurement of accuracy was easily obtained.

The Weighted Overhead Task was serves a simulation of overhead work involved heavy objects. The subject was to align a weighted box in the cavity positioned overhead. The subject further holds the weighted box overhead in that position as long as they possibly can without moving the box from the cavity. Figure 11 shows a subject holding the weighted overhead box during the weighted overhead task.

3.3 Performance Parameters

The performance parameters served as the base of the experimental evaluation. During the actual process simulations, the tile technicians' efforts were timed, observed, and video taped for further study. The parameters measured were task completion time during processes P-601 and P-301, endurance time during Weighted Overhead Task, task accuracy during Rotary Pursuit Tracking, and body discomfort throughout the experiment.

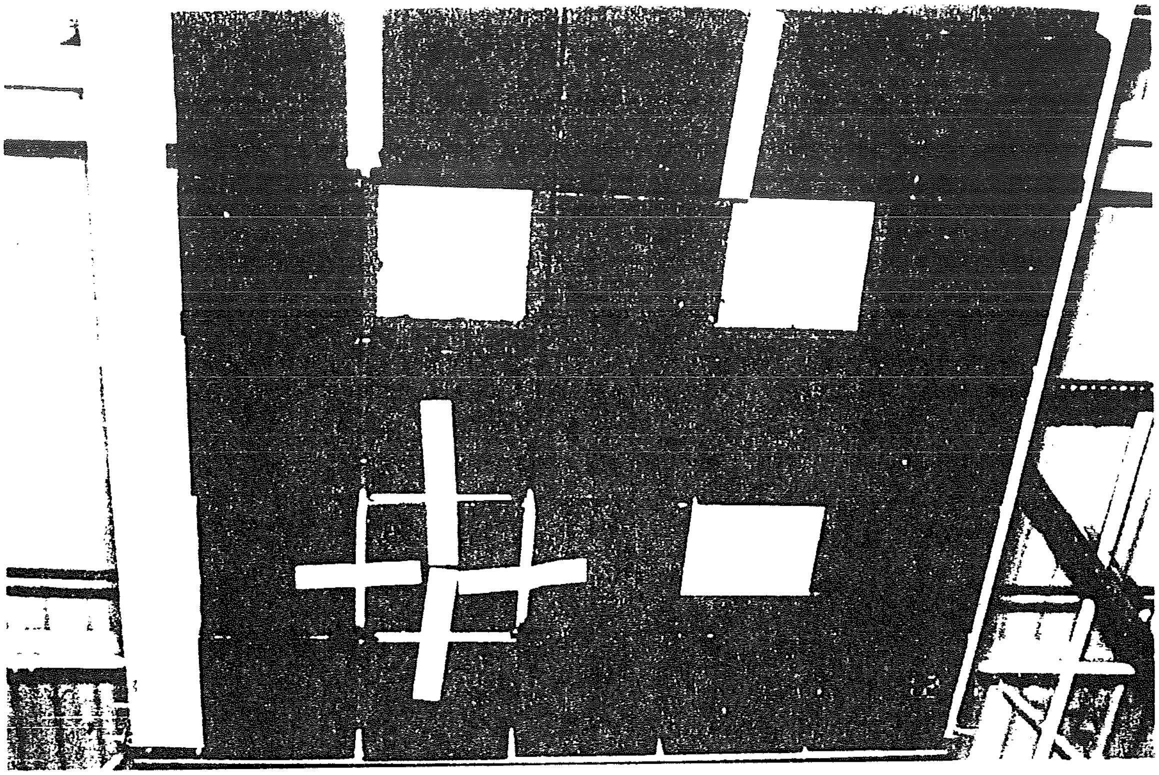


Figure 5. Simulated Orbiter Surface

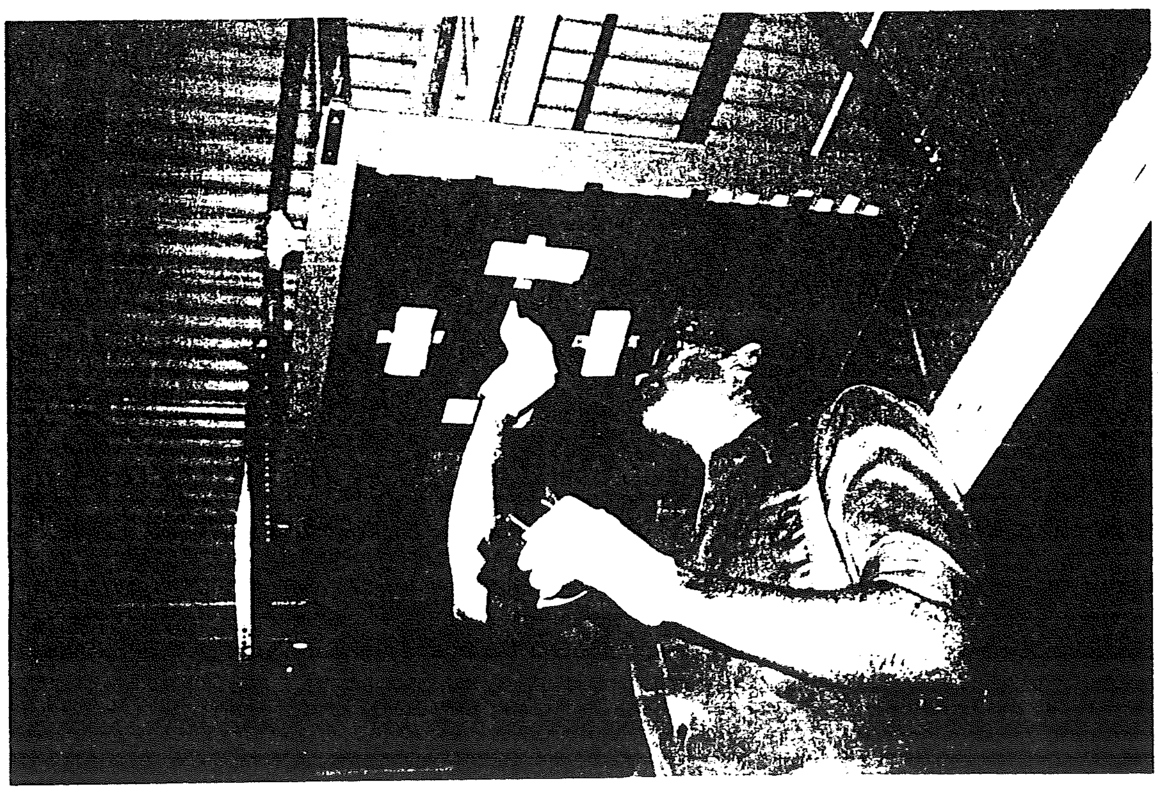


Figure 6. Subject Performing P-601

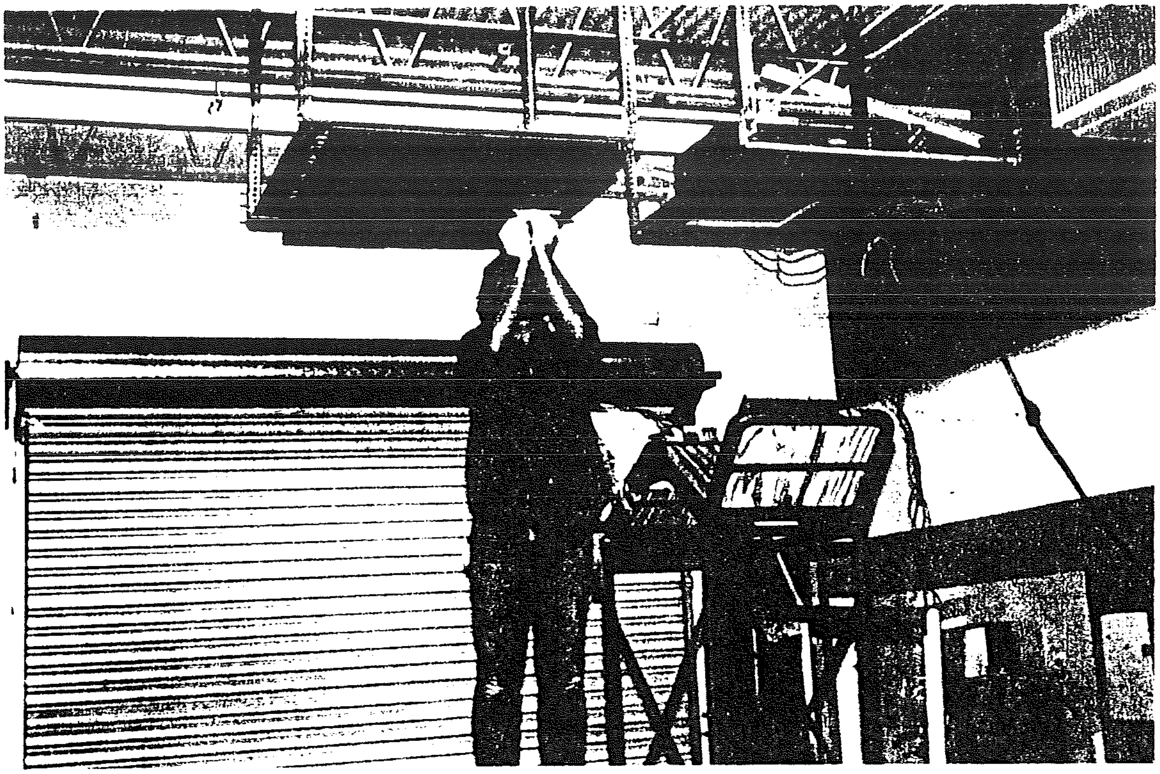


Figure 7. Subject Performing P-301

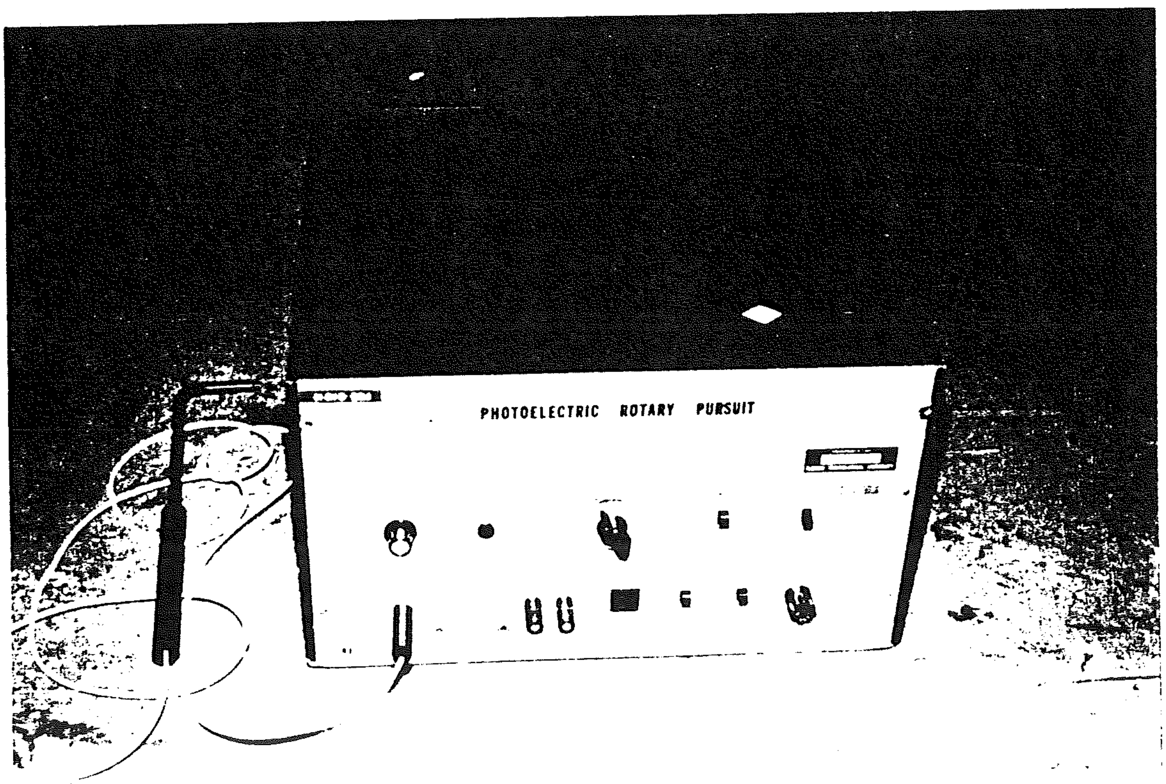


Figure 8. Photoelectric Rotary Pursuit and Stylus

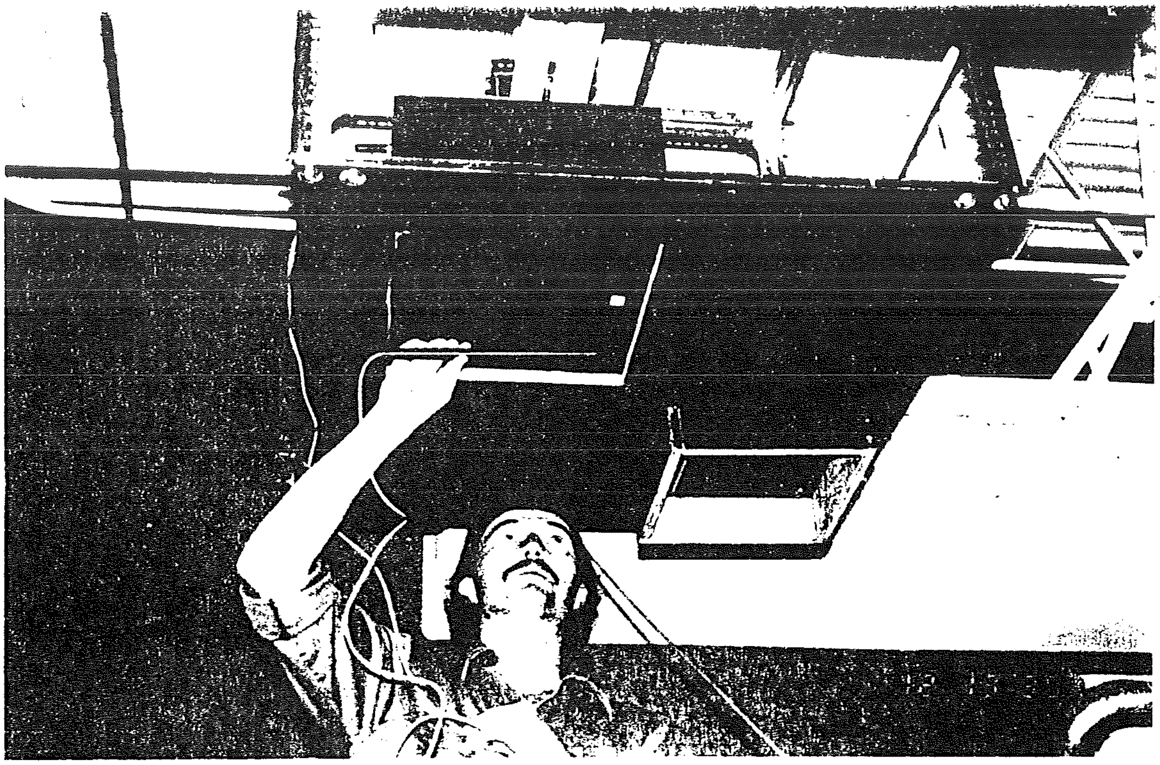


Figure 9. Subject Performing Tracking Task

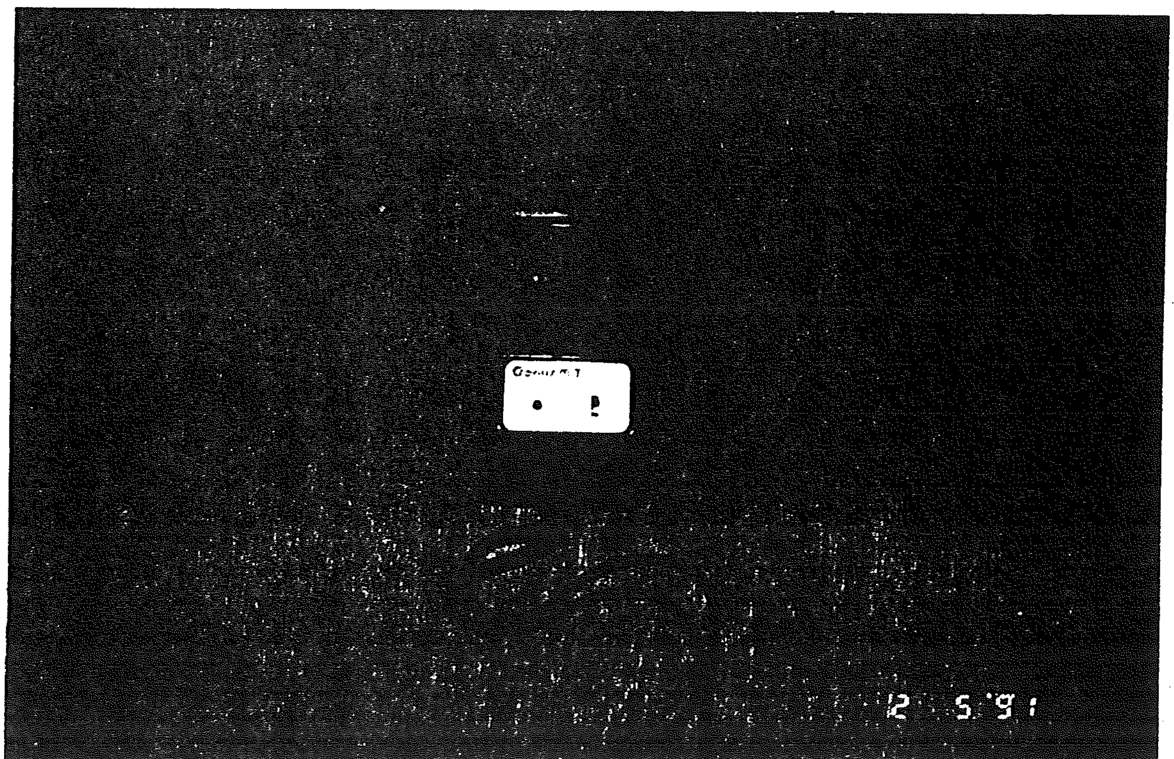


Figure 10. Digital Stop Clock

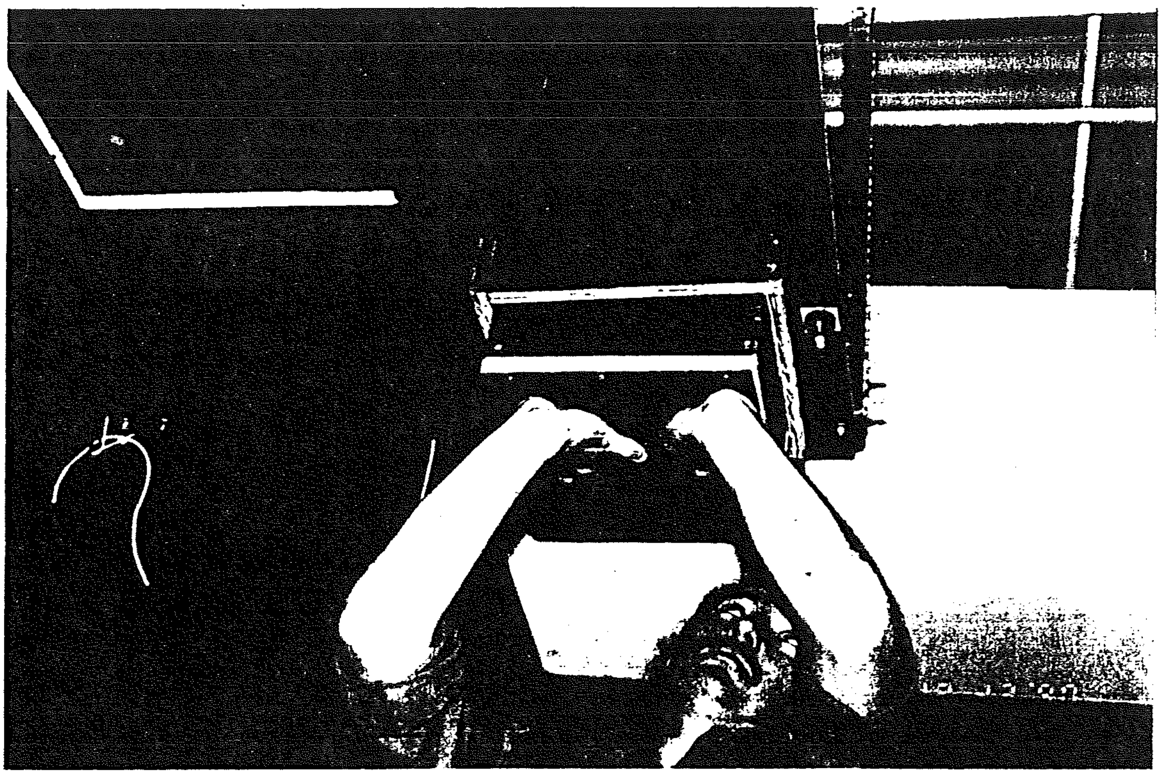


Figure 11. Subject Performing Weighted Overhead Task

3.4 Experimental Procedure

During the experiment the performance and body discomfort of tile technicians were measured in an effort to contrast the level of performance and comfort associated with both the typical standing overhead working position and the semi-reclined and supported position offered by the ergonomic seat.

Five subjects were recruited from the tile technician population at KSC. They were fully instructed as to the nature and purpose of the experimental evaluation and encouraged to perform the processes at their usual level of effort and rate. The technicians were satisfactorily compensated for their time and efforts.

The experiment involved five subjects and four processes; two tile repair and maintenance simulations and two general application tasks. The whole experiment was replicated once in order to reduce the experimental error.The subjects performed the four processes in two working positions; standing and seated. The experiment required four separate visits from each subject for a time duration of approximately 1.5-2 hours each. Each visit was virtually the same except for the position in which the processes were completed. Each subject completed all four processes on each visit. During the performance of each process, the work performance and body discomfort of each subject was measured. The parameters initially measured were task completion time, endurance time, task accuracy, and body discomfort.

The experimental procedure was as follows:

1. During the initial visit, the full objectives of the study and the experimental procedures were presented to the subjects in detail.
2. Randomly choose the overhead working position for each experimental run, standing or seated.
3. Instruct the subject to obtain the necessary amount of RTV for P-601, Installation of Pillow (Alumina) Gap Fillers from the laboratory mixing area.
4. Instruct the subject when to begin performing the first process, P-601 . Simultaneously, time, video tape, and observe the technician throughout the process.
5. Instruct the subject to complete the comfort survey upon final completion of P-301.
6. The subject will be asked to obtain the necessary amount of RTV for the second process P-301, Tile Installation.
7. Instruct the subject when to begin performing the second process, P-301 . Simultaneously, time, video tape, and observe the subject throughout the process.
8. Again, instruct the subject to complete the comfort survey upon final completion of P-301.
9. Instruct the subject to practice on the Rotary Pursuit Task for 5 minutes.
10. Instruct the subject to rest for 5 minutes.
11. Instruct the subject when to begin performing the Rotary Pursuit Task.
12. Instruct the subject to stop the pursuit task after 10 minutes. Record the readings on the time-on-target clock and the revolution counter.

13. Again, ask the subject to complete the comfort survey.
14. Instruct the subject to rest for 5 minutes.
15. Instruct the subject when to pick up the weighted box. Record the amount of time it takes to position the box within the cavity.
16. Carefully monitor the light indicator for contact. Record the endurance time of the subject.
17. Ask the subject to again complete a comfort survey.
18. The subject will asked for any comments upon completion of the entire experimental evaluation is over.

4.0 Results and Findings

The results of the experimental evaluation revealed that ergonomically designed workseat improves the work performance and provide body comfort during overhead operations especially for long duration activities. More specifically, the following summarize the evaluation results.

1. The seat dimensions were suitable for the conduct of the majority of operations except the lack of head and neck mobility offered by the workseat. However, the original workseat design include the adjustable headrest which was not implemented in the prototype due to cost constraints.

2. The worker's endurance time was significantly increased when the workseat was used. There was an average of 63 percent increase in endurance time as shown in Figure 12.

3. Task accuracy was significantly increased as shown in Figure 13 when the workseat was used. This reflects directly on the quality of work.

4. There was no significant difference, with and without workseat in the task completion time. This result is shown in Figure 14. This may be due to relatively short time duration required to complete the experimental tasks. However, when coupled with the large differences in endurance time, there will be a significant difference in task completion times over the course of long work hours.

5. The workseat greatly reduced worker's body discomfort caused by overhead positions. In some specific body parts such as lower legs and thigh, the body discomfort was eliminated. Specific body discomfort associated with each posture is shown in Figure 15.

Based on the above mentioned findings, it is concluded that the ergonomically designed workseat is effective in improving the productivity of tile technicians and in reducing their body fatigue.

The tile technicians' general comments were also compiled during the use of workseat. The following are some comments, which would affect the prototype buildup, shared by most of tile technicians during the use of workseat.

1. There was not enough clean work area, tools and equipment storage, and waste container. A tool "tray" or some other means for the operator to have access to the tools without changing position is necessary.

2. The technicians concerned over dripping RTV and other chemicals as well as falling debris. A protection such as a face shield is desirable.

3. There is a need for information access while work is being done for some operations as well as exchanging information with floor crew.

(4) There is a need for an arm rest in each side.

The above mentioned comments and suggestions were implemented in second detailed workstand design.

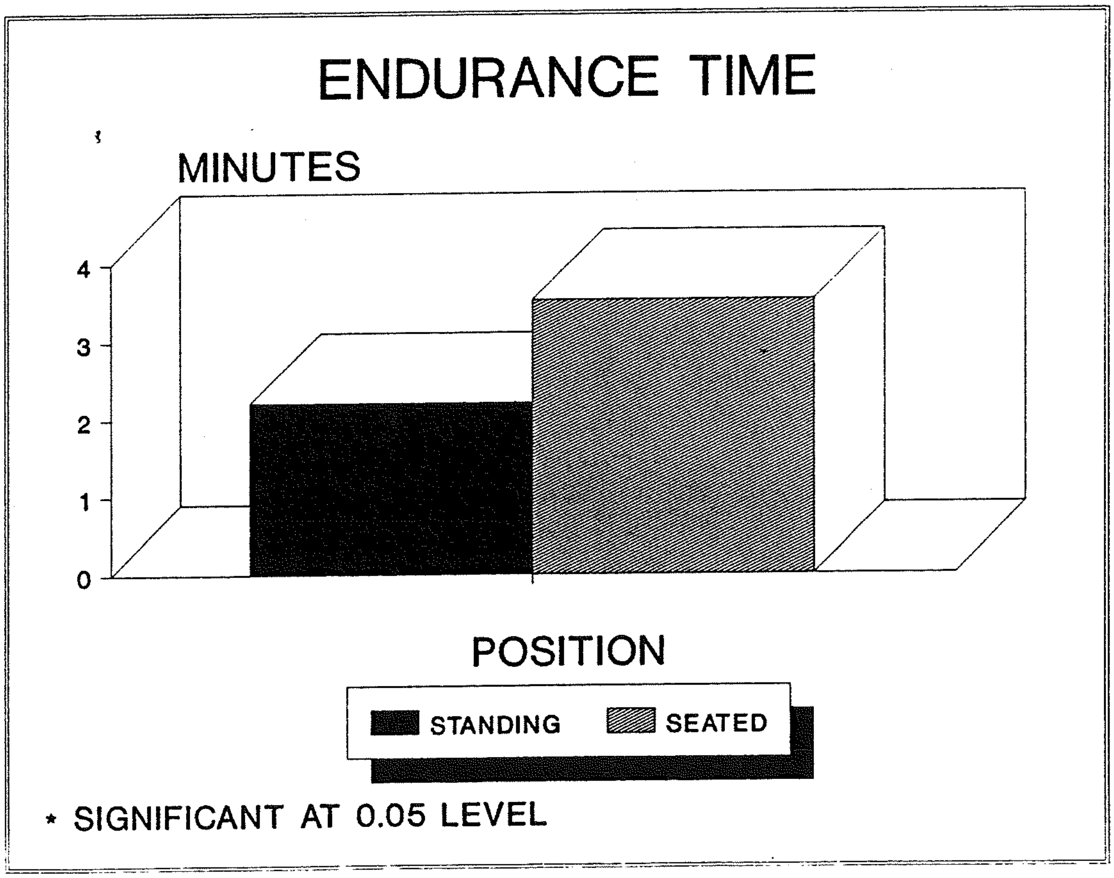


Figure 12. Endurance Time

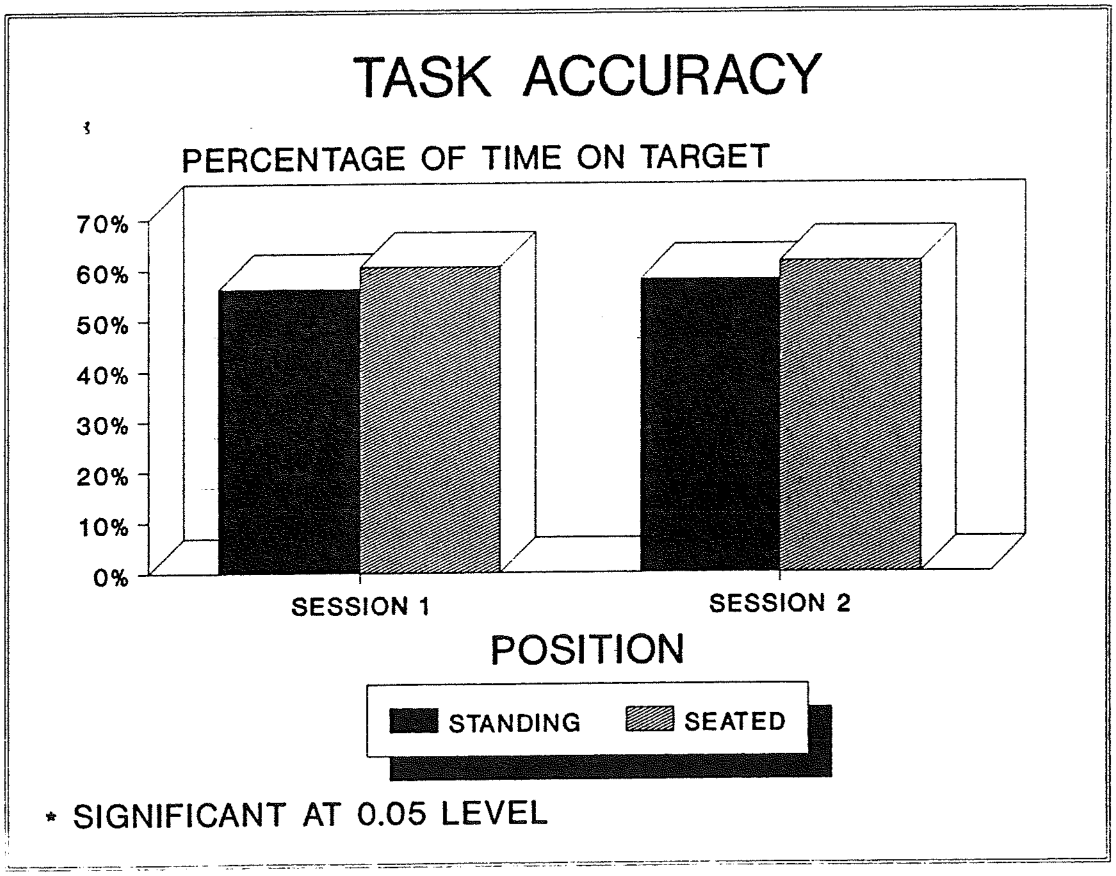


Figure 13. Task Accuracy

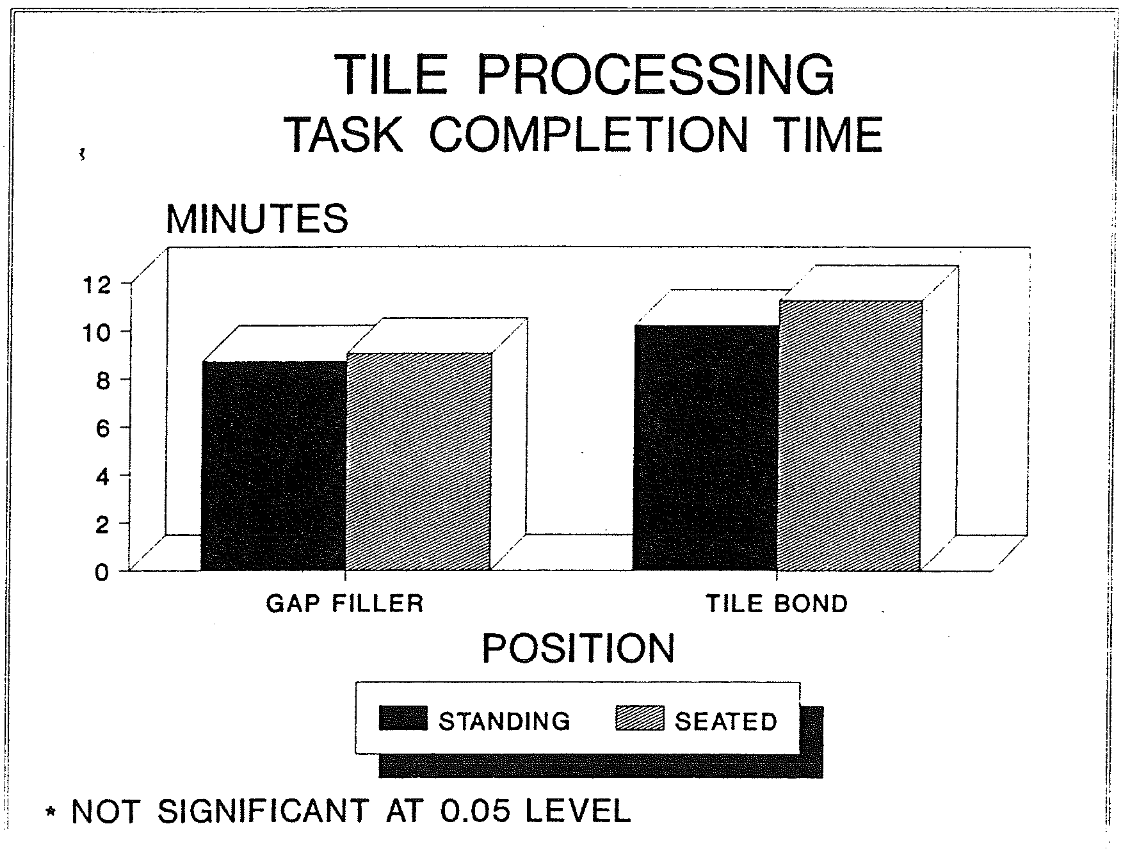
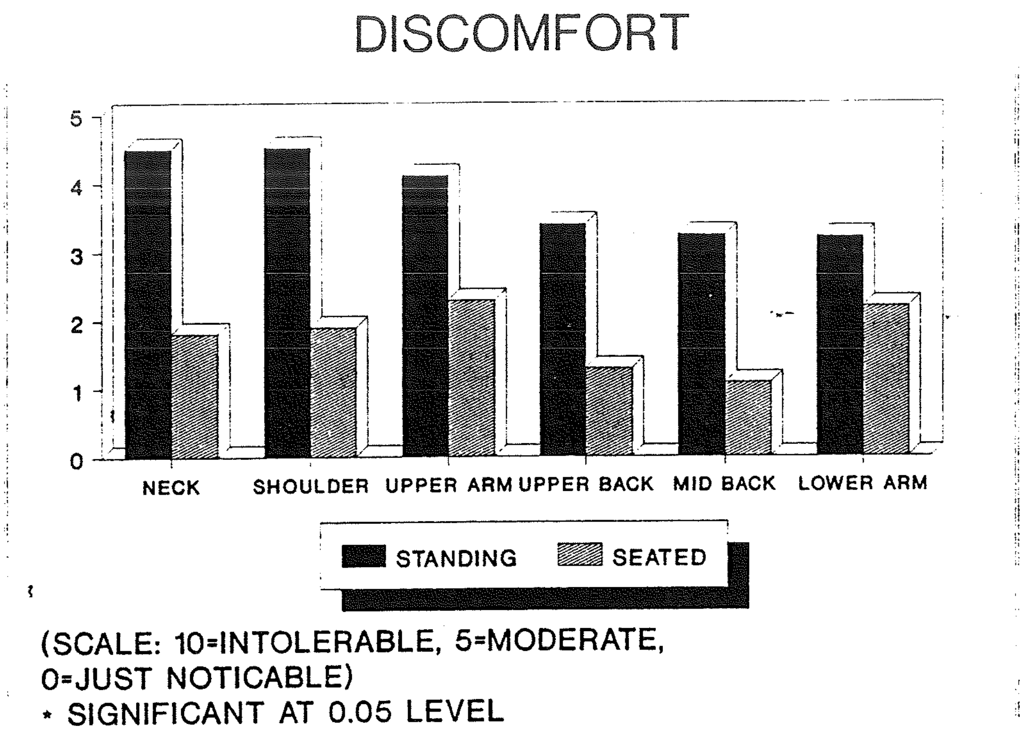


Figure 14. Task Completion Time



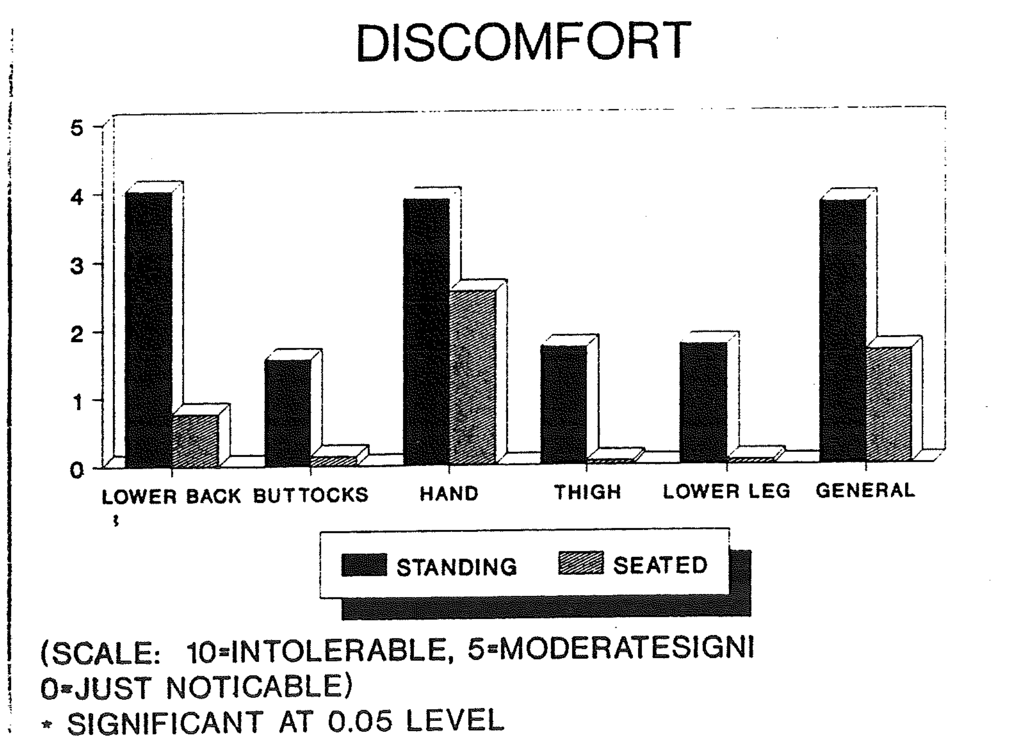


Figure 15. Body Part Discomfort

Appendix II

Voice Data Entry

UNIVERSITY OF CENTRAL FLORIDA/KSC
COOPERATIVE AGREEMENT

PROJECT #6

PRODUCTIVITY TECHNIQUES

VOICE DATA ENTRY COMPONENT

PHASE II - REPORT

SPRING 1991

TABLE OF CONTENTS

1.0 INTRODUCTION

Voice Data Entry (VDE) is an evolving technology that converts natural speech to computer data through voice recognition components and computer data to speech/voice output through a speech synthesis component. A typical voice recognition system allows the user to enter data, issue commands, and perform transactions by verbally entering information using natural, conversational speech. Manufacturers of state-of-the-art recognition hardware claim that vocabularies of hundreds of words can be recognized with recognition rates of 99 percent. The recognition process may or may not be user dependent. Speech synthesis could allow a computer to verbally prompt a user during an application procedure.

Previous efforts with respect to tile processing include a pilot project by Lockheed Space Systems and Stanford University, California, through the SIORA (Space Integration & Operations Research Applications) program that allowed technicians to verbally enter numeric step and gap measurements as they pertain to P-310 (tile step and gap measurement). The University of Central Florida, through the PHASE I COOP agreement program, developed the training and implementation strategy for the Lockheed pilot project (refer to PHASE I interim report). The current objectives as stated in February 1990 statement of work include:

1. STUDY THE FEASIBILITY OF A MICROCOMPUTER BASED VDE SYSTEM -- Problems identified in the step and gap application indicated that the step and gap process was not the most viable candidate for VDE. Advances in the field of microcomputers indicate that it may be feasible to use a microcomputer based system. This use will be investigated by UCF.

2. DESIGN AND CARRY ON AN EXPERIMENT -- Through experimentation, we can make an assessment of the feasibility of using microcomputer based speech recognition technology in orbiter processing activities.

3. STUDY POSSIBLE PROBLEMS IN INTERFACING WITH KSC'S COMPUTER SYSTEM (i.e., SPDMS II) -- KSC is currently developing/converting many of its systems to be computer based. SPDMS II (Orbiter Processing Data Management System II) will be the computer platform for these efforts. It is envisioned that all systems deployed at KSC will share this platform. If VDE becomes feasible as a computer communication medium, it is important to study any interfacing problems with SPDMS II.

4. EXPLORE NEW APPLICATIONS OF VDE -- UCF is striving to obtain a familiarity with VDE that can only come from a prototyping effort. While an initial effort may center on the tile processing environment, we also recognize that the TPS is only a fraction of the systems involved with orbiter processing. Future endeavor's may extend to other facets of orbiter processing. A systematic procedure for identifying processes amiable to VDE will be developed.

2.0 KSC POTENTIAL APPLICATIONS

2.1 GENERAL

Following are some of the job characteristics which would qualify a job to be considered for VDE:

- TASKS REQUIRING COMMUNICATION WITH A COMPUTER - Since VDE is a means of communicating with the computer, any job which requires such communication as part of its procedure may be considered for VDE application. Other factors such as the environment, the speed of communication, and the reliability need to be assessed as well.
- FREE HAND ENVIRONMENT REQUIRED OR DESIRABLE - When a job requires that manual labor of some sort be interrupted to write down or key in data, then VDE may be considered as an alternate form of computer interaction.
- DATA ENTRY REQUIRED IN PERFORMANCE OF TASK - When large amounts of repetitive data are to be entered to a data base, a reliable voice data entry system can make substantial savings in terms of time thus resulting in an increase in productivity. Benefits may be even greater if the operator is not skilled in the use of the keyboard or that the keyboard is not designed or positioned properly for data entry.
- A LIMITED AND CONSISTENT JOB TASK VOCABULARY - Voice data entry systems are more reliable and efficient if they deal with limited vocabularies. Jobs with limited vocabularies, or those that can be redesigned to use a limited vocabulary may be eligible for VDE.
- STABLE MANPOWER, AS VOICE RECOGNITION REQUIRES TRAINING - Studies have shown that training the system for specific users can be a limiting factor in the VDE application. Once the system is trained for a specific user, it is in fact faster and more reliable than the user independent systems.

A general assessment of the tile processing environment was accomplished via direct observation of orbiter processing operations, and interviews with OPF personnel. The results of this initial assessment indicate that three areas exist that could benefit from VDE technology. They are:

- STEP AND GAP MEASUREMENT (P-310)
- TILE IDENTIFICATION
- PROBLEM REPORTING

It should be pointed out here that the effort of identifying jobs at KSC which may benefit from VDE is a continuous task along this project. Efforts may be extended toward redesigning some of the identified jobs to eliminate redundancies, thus making them compatible with VDE. The three areas identified here were determined based on the general characteristics listed above and were meant to provide us with a specimen for a

laboratory experiment for a typical application at KSC.

2.2 P-310 (Step and Gap Measurement)

Although no effort was expended on further investigation of P-310, as previous efforts from other agencies have documented VDE's applicability in that area, several points are worth mentioning as to the problems associated with the P-310 VDE pilot project:

CONNECTION TO VOICE RECOGNIZER — Users communicated with the International Voice Products recognizer via hardwired connection. This restricted the user to a limited radius around the recognition hardware, in addition to possibly violating KSC policy with respect to cables in the OPF.

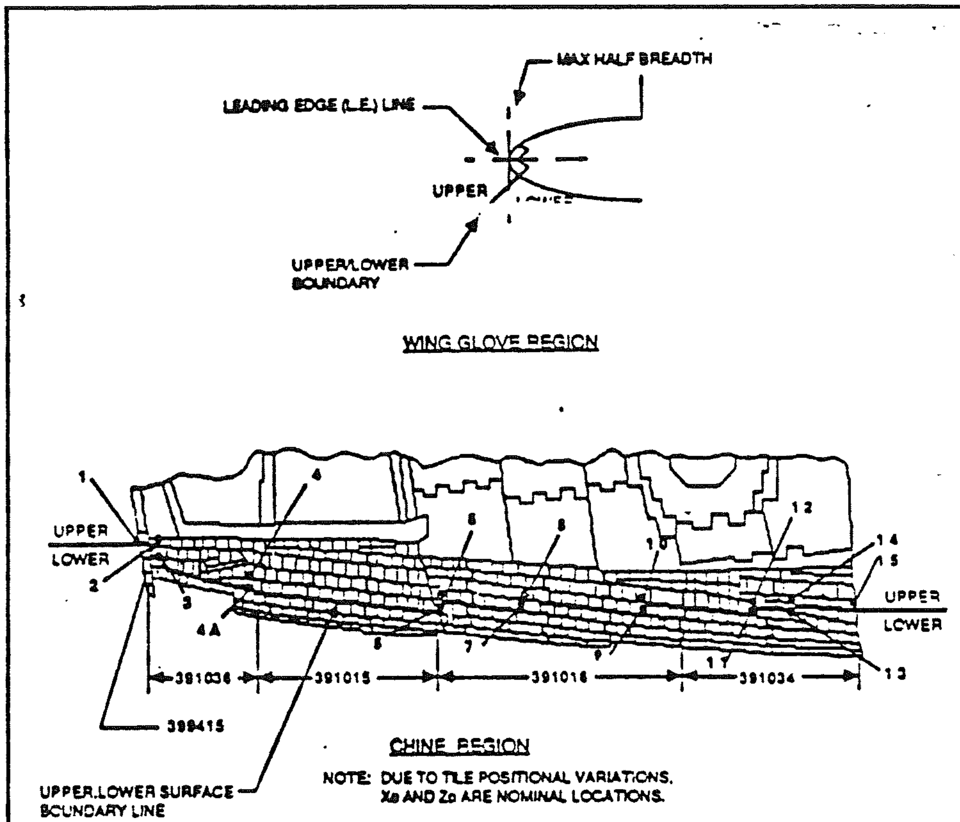
TRAINING TIME — While essentially all voice recognizers must be trained to recognize the users voice, questions were raised concerning the amount of training time necessary compared to the tile technician turn-over rate. With respect to system acceptability, some resistance by technicians and supervisors to the system has been noticed. Consideration to human engineering and the technician/supervisor acceptability to new technology is warranted for successful application.

COMPUTER PLATFORM — The SIORA pilot project was based on an International Voice Products recognizer connected to a MicroVAX located in the mezzanine between the two hi-bays. This system is not supported in the OPF environment. Future VDE configurations would have to interface with the SPDMS II platform which will be based on the IBM PS/2.

LASER STEP AND GAP TOOL — Recording measurements by VDE did not eliminate the most prevalent failure mode involved with P-310 -- the actual measurement itself. The laser step and gap tool can accurately record measurements, then send them directly to a database.

2.3 TILE IDENTIFICATION

2.3.1 CURRENT PROCEDURE — When inspecting or repairing tiles, it is critical to determine in which region a tile belongs. The orbiter is divided into several regions as shown in figure 1. Tile identification is a real concern to NASA and Lockheed engineers who must determine a tile location on the orbiter.



PT	Xo	Zo	PT	Xo	Zo
1	259.14	325.34	8	440.00	306.89
2	258.01	325.88	9	485.00	301.78
3	298.08	320.73	10	485.00	307.[illegible]6
4	310.27	312.03	11	534.50	302.37
4A	310.27	311.54	12	534.50	307.[illegible]
5	397.00	302.75	13	554.00	303.97
6	397.00	307.97	14	554.00	308.50
7	440.00	301.55	15	578.44	303.70

Figure 1 – Sample region of orbiter surface.

The current process involves visual reference to drawings to determine a tile's region. While tiles toward the center of the region are easy to identify, tiles which lie on or near the border between two regions may cause difficulty. If the technician cannot clearly discern to which region a tile belongs, he must consult with a Rockwell engineer who in turn consults a computer database which contains location data on each tile. The process relies heavily on technician experience and familiarity with tile regions. Moreover, this task can be a frustrating and time consuming process and subject to error. For better quality of work and reliable identification, consultation with the data base may be essential while the technician or engineer is in a remote location with respect to the computer terminal. A VDE system may be of value in such situations.

2.3.2 PROPOSED PROCEDURE — The proposed procedure would connect the engineer directly to a tile region database. If consultation is to occur, the engineer need only speak the tiles identification number and be given a voice synthesized computer response consisting of the tiles location, along with any other pertinent data that could be supplied by the database. One advantage for such an application is that the vocabulary to be used may be limited to character by character voice data entry, which may be user independent (i.e., no training for system is required).

2.3.3 EXPECTED GAIN IN PRODUCTIVITY — Although time savings could be realized in certain cases, indirect gains would be immediately obtained in terms of quality of data such as instant tile location. For proper quantification of expected gain in productivity, all jobs requiring tile identification and or data base query while not in position to use a computer terminal may have to be surveyed and the use of VDE be assessed and compared to the current procedure.

2.4 PROBLEM REPORTING

2.4.1 CURRENT SYSTEM

Problem Reporting (PR) with respect to the TPS is a procedure to document problems identified with the orbiter's protective tiles. Examples of problems include a tile that is chipped, gouged, debonded, or missing. These are problems that will require corrective action. The PR documents the exact nature of the discrepancy, and becomes the cover sheet for a Work Authorizing Document (WAD) that will eventually include the engineers disposition report, which is a detailed description of the steps needed to correct the problem.

The basic flow of problem reporting and the PR document production can be summarized as seen in figure 2 below:

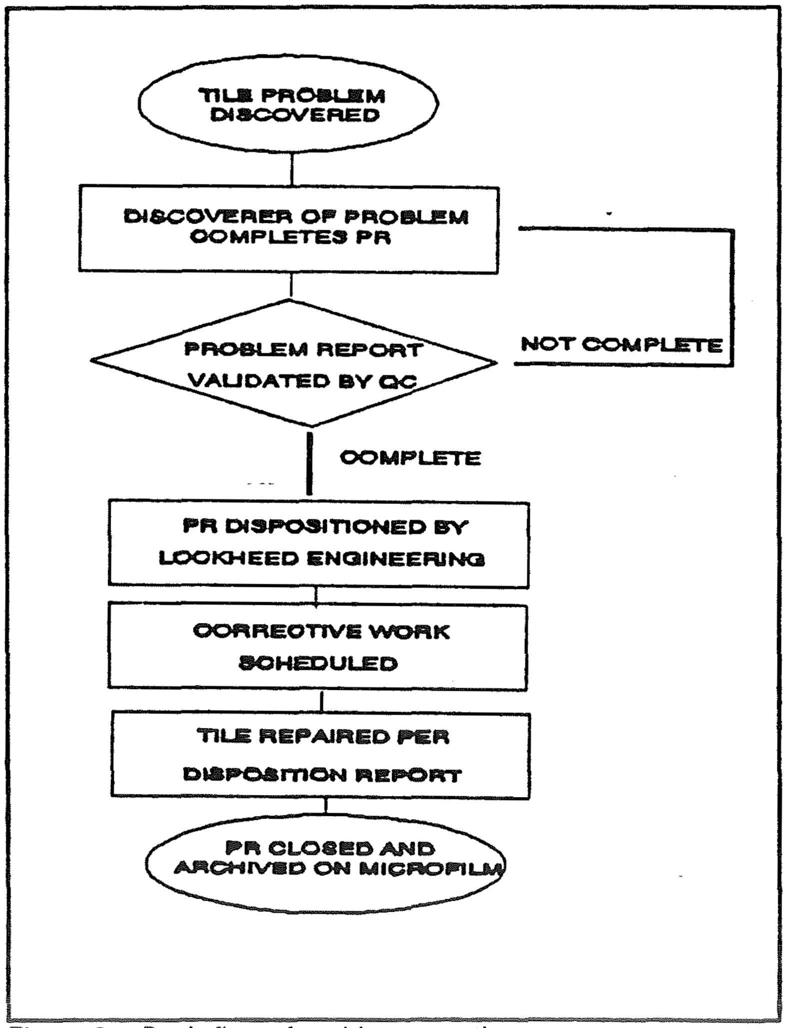


Figure 2 -- Basic flow of problem reporting.

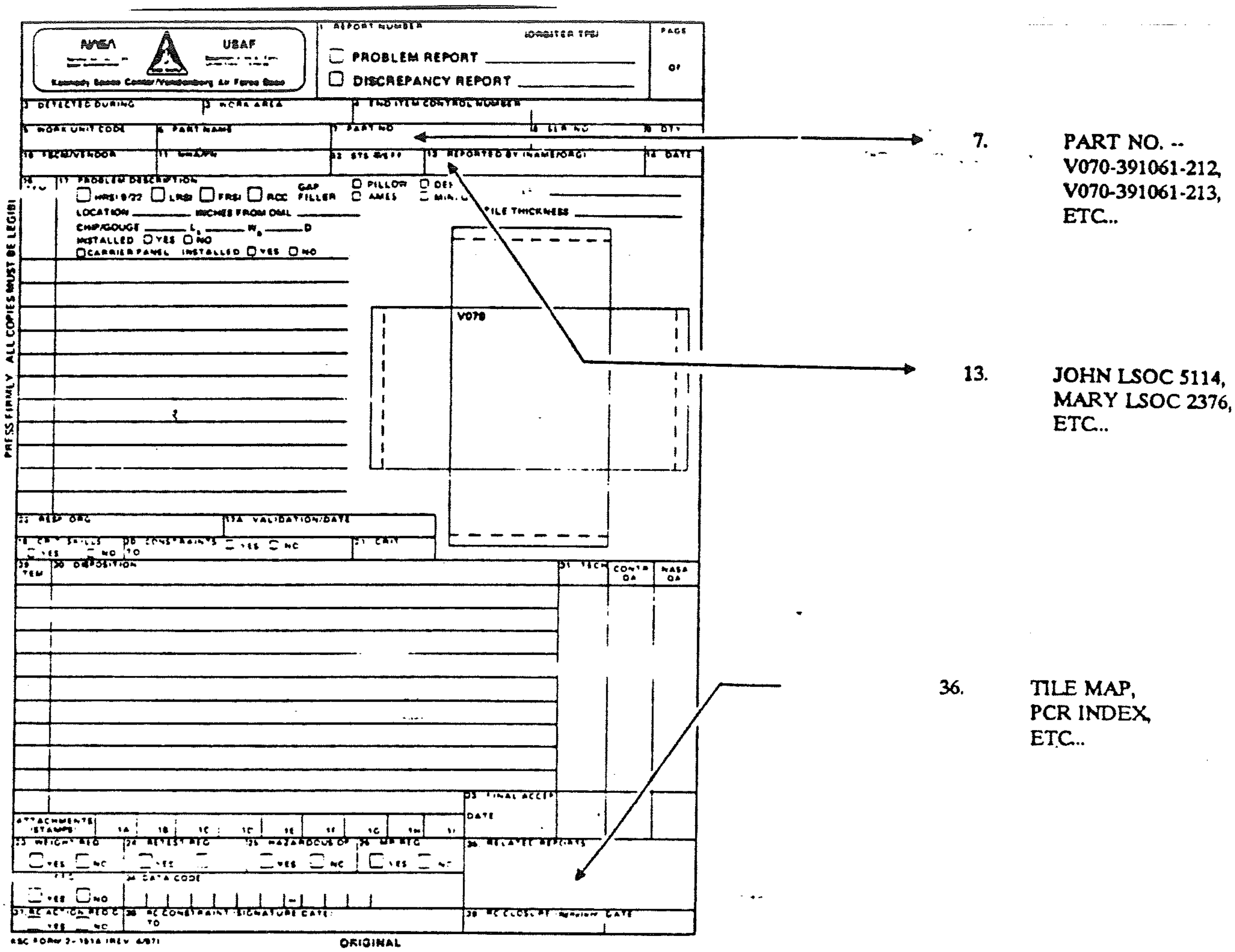
2.4.1 CURRENT SYSTEM (Cont.)

Problems on the orbiter can be identified by any technician. Usually, a team of quality supervisors inspect the orbiter upon landing at KSC. The process is usually done in the OPF environment, however a pre-inspection may take place in California if the orbiter lands there. Each QC inspector (technician) is responsible for ceratin section of the orbiter. The actual process of identifying anomalies is done by observation. Once a problem is identified, the inspector completes a problem report form (see figure 3). In a typical process, the inspector may have to consult technical documentation or computer data bases for proper completion of the report. Some of the "vocabulary" used in the report is standard such as tile number, while other areas are not standard, especially the problem assessment. It is believed, though, that there are enough problem reports available on microfilm which if compiled may produce a standard vocabulary to cover the majority of the problems. The PR is then validated by a QC inspector who ensures that the report is complete and has enough information to assess the problem. The PR is then dispositioned by a Lockheed engineer, who details the work required, thus producing a Work Authorizing Document (WAD). Corrective work is then scheduled in such a way as not to conflict with other maintenance work be performed. Corrective work is then performed according to the directions given in the WAD. Corrective work performed by the technician is then "bought off" by the quality supervisor. Finally, the PR is signed by representatives from all affected areas and the completed WAD, which can be quite bulky at this point, is sent for archiving on microfilm.

2.4.2 PROPOSED SYSTEM -- PROBLEM REPORTING WITH VDE

Developing the necessary grammar definition file to recognize PR data would be facilitated by the programming structure and features of a voice development software. By categorizing each block of the PR as its own grammar, the possible responses can be limited to only that which is legal within that particular block. A typical TPS PR, along with its sample grammar definition file, is shown in figure 3 to illustrate this point.

2.4.2 PROPOSED SYSTEM -- PROBLEM REPORTING WITH VDE (Cont.)



With the exception of blocks 17 and 30, all blocks have a relatively limited vocabulary that can be assumed to be well within the capabilities of any VDE system. Blocks 17 and 30, however, involve describing the exact nature of the problem in engineering terminology. This presents a challenge to developing a VDE application for problem reporting. The possible responses when describing tile damage are numerous, however a modified PR procedure may limit the vocabulary used and put it within current VDE capability. A more thorough study is essential for this application.

2.4.3 PROBLEM REPORTING WITH VDE

In trying to determine the feasibility of this evolving technology as it applies to tile processing, it becomes necessary to consider the functional areas involved and how they can integrate with VDE technology.

The functional areas that have been identified with respect to tile processing are:

- RELIABILITY - Errors in PR's can be time consuming and costly, particularly when the wrong tile is worked or pulled. Several failure modes for tile processing have been identified. They are:

  - TECHNICIAN MISIDENTIFIES ANOMALOUS TILE -- EFFECT: POSSIBLE REPAIR OF WRONG TILE.

  - TECHNICIAN MISIDENTIFIES ANOMALOUS TILE, WRITES UP PR WITH INCORRECT TILE ID, AND IMMEDIATELY REPAIRS DAMAGED TILE -- EFFECT: LOSS OF TRACEABILITY

  - TECHNICIAN INCORRECTLY TRANSPOSES PR DATA FROM ONE FORM TO THE NEXT DUE TO THE ILLEGIBILITY OF THE ORIGINAL PR -- EFFECT: A HOST OF POSSIBLE ERRORS, SUCH AS DELAY OF WORK, INCORRECT REPAIR, ETC..

  VDE can reduce the errors associated with tile identification and recording by providing the technician with the ability to simultaneously view and report the damage on the anomalous tile. Data entered by voice can then be instantly verified through synthesized voice response.

- HUMAN FACTORS - SPDMS II calls for PR data to be entered into a data base via IBM PS/2's acting as terminals for the mainframe computer. The primary means for communicating with computers is through the keyboard. This requires a certain level of typing proficiency if the data is to be entered in a timely and accurate manner. VDE provides an alternative to the keyboard by allowing spoken phrases to be sent to the computer just as if they had been typed on the keyboard. Further, the keyboard requires that the technician be seated in front of the terminal when entering data. This offers little improvement over the manual method as technicians must still record data manually when under the orbiter and then return to the terminal to input the data. A wireless voice link with the computer would eliminate the interim step of recording data from the field so as to return to the terminal. Technicians could enter the data while looking at the problem under the orbiter.

- CONFIGURATION MANAGEMENT - Tile locations on the orbiter are maintained via drawings (or most recently via CAD software) that the technician must refer to if in doubt about the identity of a certain tile. Misidentification of tiles could be further avoided if an on-line expert system were available to assist with, and/or verify, tile identification.

3.0 INTERFACING WITH KSC's COMPUTER SYSTEM

3.1 SPDMS II

SPDMS II, Orbiter Processing Management System II, is a new computer work control system that is being implemented by Lockheed. It will control the Orbiter processing operations by:

- CONTROLLING WORK PACKAGE PRODUCTION AT THE TEST, ASSEMBLY, AND INSPECTION RECORD (TAIR) STATIONS
- CONTROLLING FACILITY RESOURCES (TECHNICIANS, ENGINEERS, SAFETY PERSONNEL, MATERIALS, AND EQUIPMENT)
- CONTROLLING PROBLEM REPORT (PR), INTERIM PROBLEM REPORT (IPR), TEST PREPARATION SHEET (TPS), AND DEVIATION (DEV) PROCESSING

With respect to the third bullet, that of controlling problem reports, VDE (Voice Data Entry) can have a positive effect on interactions between technicians and computers.

SPDMS II, which has yet to see full implementation, will improve orbiter processing efficiency by:

— RELEASING ONLY "WORKABLE" PACKAGES

— DECREASING PAPERWORK

— IMPROVING ACCURACY OF RECORDS

— IMPROVING WORK SCHEDULING

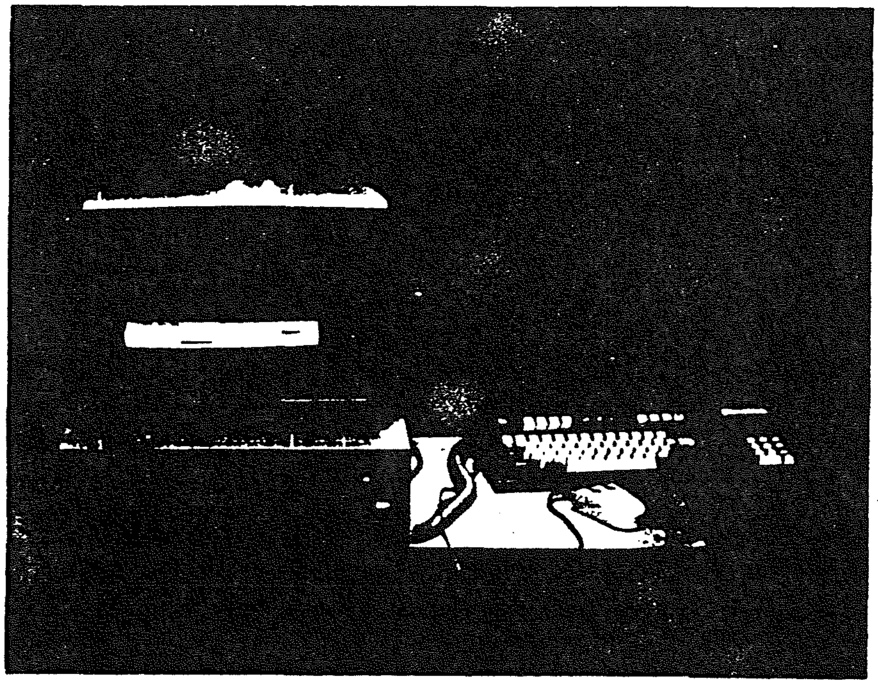
— IMPROVE WORK ACCOUNTABILITY

— PROVIDING NEAR REAL-TIME STATUS

VDE can impact these goals by offering an alternative means of interacting with the computer. Since SPDMS II will involve installing computers on the OPF floor, VDE can play a role in enhancing the man and machine interaction.

3.1 SPDMS II (Cont.)

SPDMS II will be comprised of computer workstations located in all areas of KSC including the HI-BAY floor. These workstations will be composed of IBM 7561 industrial version PS/2's operating under an OS/2 multitasking environment. These computers will have 105 Megabyte hard drives, and 16 Megabytes of RAM. Several computers will be networked together via token ring, each having the ability to interface with NASA's IBM 3090 mainframe computer. The IBM 7561 PS/2 is pictured below:



3.2 INTERFACING VDE WITH SPDMS II

The Verbex voice recognizer, can be interfaced to NASA's proposed SPDMS II computer system. The majority if VDE systems can easily be interfaced with SPDMS II's network. The recognition and synthesis process required in any application can be handled at the workstation level.

Fortunately, the Verbex hardware can be connected to the above system with little or no modification. For most applications, such as problem reporting, the Verbex would be communicating directly with the PS/2 which would be running the problem report module software. The verbex can, however, communicate with the IBM 3090 as well due to the fact that both the Verbex and the PS/2's use VT-100 terminal emulation. This was proven in the UCF laboratory as explained later. As currently envisioned, the Verbex would be connected to the PS/2's via COM-Port cable.

In this configuration, the Verbex functions as an extension of the keyboard, allowing the user to send data by "speaking" to the computer via headset, just as if he were typing the data manually.

The Verbex receives voice patterns from the user by a microphone which can be hardwired to the unit or sent by radio waves from a wireless transmission unit. The hardwired approach produces the greatest accuracy, but it requires the user be seated in front of the computer when speaking as shown below:

The wireless approach, which is still under study for feasibility, outfits the user with a headset and a portable transmitter/receiver which allows the user to converse with the computer hands free. While this approach allows for the greatest mobility of the user, interference from metal scaffolding, as well as DOD limitations on transmitters within the OPF, may prohibit its use in the current context of applicability.

Page intentionally left blank

4.0 EXPERIMENTAL EVALUATION OF VDE HARDWARE

4.1 INTRODUCTION:

UCF has acquired the VERBEX SERIES 5000 voice recognizer for the purposes of determining the systems general capabilities. Since the operational environment at NASA presents special problems to the wireless voice recognition process, an experiment was conducted in the UCF lab to determine a baseline for operational characteristics of the VERBEX system.

UCF's VDE work is oriented towards determining OPF tasks compatible with VDE. An element of this compatibility is that of a hands-free working environment.

4.2 THE PROBLEM:

Future evaluation efforts of potential KSC tasks will require a knowledge of the capabilities/limitations of the current state-of-the-art voice recognition equipment.

One of the task characteristics identified as making a job compatible with VDE is that of a hands-free working environment. While a hardwired approach can attain this goal, the user is restricted to an area around the hardware not exceeding the length of the cable. The user can be freed from this restriction if a wireless approach is utilized. This would entail outfitting the user with a portable transmitter/receiver that would communicate with a base unit located next to the recognition hardware. To test the feasibility of this approach, UCF has also procured the COMMUNICATIONS APPLIED TECHNOLOGY RS111 base radio assembly and the QB-1C mobile transmitter/receiver for testing purposes with the VERBEX SERIES 5000 recognizer.

4.3 EQUIPMENT NEEDED FOR PROPOSED RESEARCH:

Previous efforts with respect to VDE at KSC were based on the International Voice Products continuous voice recognizer. This system served as the platform for the P-310 voice application prototype. While it performed adequately, doubts about its ability to handle more complex applications were raised in light of its limited memory (100 words in a single application only). UCF choose not to select the same system but rather a more powerful one with higher vocabulary capacity.

The VERBEX SERIES 5000 is a continuous speech recognition system from Verbex Voice Systems, Inc.. The model that UCF acquired has a theoretical upper limit of 600 to 800 words maximum, but only 80 words can be active for recognition at any one time. In order to utilize the full power of the system, multiple vocabularies must be used. Users communicate with recognizer via hardwired headset or optional radio communication device. The optional radio device is described next.

The COMMUNICATIONS APPLIED TECHNOLOGY RS111 base radio assembly and the QB-1C mobile transmitter/receiver allow the user to communicate with the recognizer in a wireless vicinity. The crystal frequency controlled system broadcasts with 50 milliwatts of power. The microphone has a range of 200 - 4000 Hz within +/- 2 Db. *(NOTE: Experiment was conducted with equipment as received with no modifications.)*

4.4 SUBJECTS:

Two subjects, both male, were used. Data collected from the two subjects was analyzed separately and collectively.

4.5 EXPERIMENTAL METHODOLOGY:

Dependent and independent variables for this experiment were identified as follows:

DEPENDENT VARIABLES:

- Phrases Recognized (after 1, 2 or 3 attempts)
- Phrases Presented

INDEPENDENT VARIABLES:

- Distances from the base receiver
- Speaker

CONTROLLED VARIABLES:

- — Obstacles between receiver and transmitter.
- — Other radio frequency emitting devices (computers)
- — Background noise
- — Vocabulary complexity
- — Number of recognizer training passes completed

The experiment that was conducted tested the operational characteristics of both the radio system and the voice recognizer. The first experiment was designed to test the responsiveness of the voice recognizer to the radio-system. This was accomplished by attempting wireless recognition at various ranges from the base unit. The following control conditions applied to the test:

- LINE OF SIGHT COMMUNICATION (NO OBSTACLES)
- TWO REPLICATIONS OF THE EXPERIMENT WITH SUBJECTS (SEPARATELY)
- LOW AND MID LEVEL VOCABULARY COMPLEXITY (SEPARATELY)
- 2 TRAINING PASSES

The following schematic indicates how the unit was tested. The recognizer and base receiver were placed at the end of a 250 foot hallway. The user then attempted recognition at the following distances: 0, 50, 125, 150 FEET. Each of these distances were tested with two vocabularies.

The first vocabulary had a complexity factor of 4%, while the second had a factor of 23%. *(Briefly, the system calculates a complexity factor for a vocabulary based on the speech data entry rate versus the data processing rate -- 100% complexity would just allow the recognizer to keep up with the speaker. Less than 100% the recognizer can more than keep up, but more than 100% the recognizer may not be able to keep up with the user in real time.)*

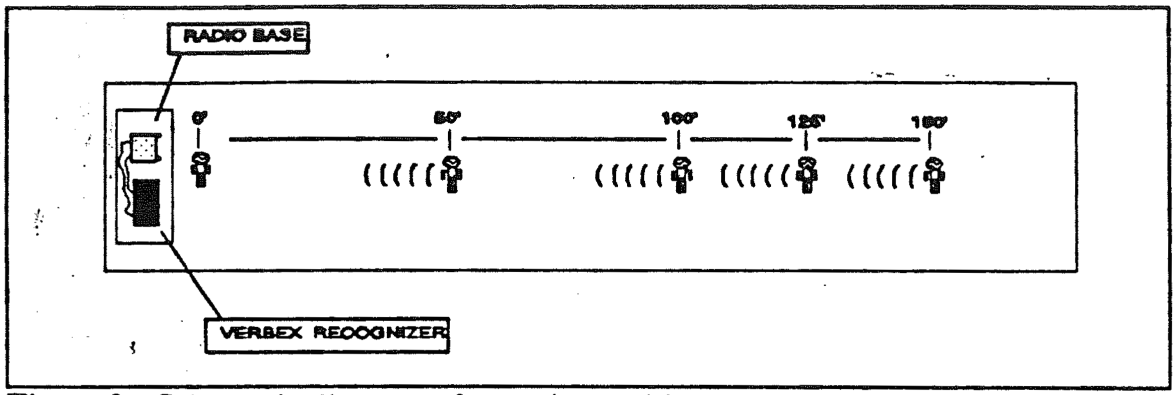


Figure 3 - Schematic diagram of experimental layout.

For each vocabulary, and at each of the specified distances, the script phrases indicated in figure 2 were spoken:

TILE IS BURNT	COATING IS BURNT
TILE IS CHIPPED	COATING IS CHIPPED
TILE IS CONTAMINATED	COATING IS CONTAMINATED
TILE IS DEBONDED	COATING IS DEBONDED
TILE IS MISSING	COATING IS MISSING

Figure 2 - Phrases to be spoken by the user.

The actual vocabularies used were designed in such a way that the only difference between them is the complexity level as perceived by the recognizer. The user attempted recognition with the same phrases as indicated in figure 2. The purposes of having a low and high complexity vocabulary to accomplish the same task is to determine how sensitive the system is to increased complexity. Both the low and high complexity vocabularies are indicated in figures 3 and 4.

```
;Vocabulary for experimental trial
!EXPVOC1=
#recognition
#grammar
.PR1

    .PR1=
     .OBJECT .PREP1 .ERROR_OF_OBJECT

    .OBJECT=
          TILE
          COATING

    .PREP1=
          IS

    .ERROR_OF_OBJECT=
          BURNT
          CHIPPED
          CONTAMINATED
          DEBONDED
          MISSING

#translation
< ||                ;no initiator string
```

Figure 3 -- 4% complexity vocabulary.

```
;Experimental vocabulary
!EXPVOC2=
#recognition
#grammar
.PR1

    .PR1=
          TILE
          COATING
          IS
          BURNT
          CHIPPED
          CONTAMINATED
          DEBONDED
          MISSING
          DISCOLORED
          CHARRED

#translation
< ||               ; no initiator string
| ||               ; no seperator
> |015              ; terminator is CR

#RES
#TE
```

Figure 4 -- 23% complexity vocabulary.

An overall recognition percentage was calculated as follows:

$$RECOGNITION\% = \frac{\sum_{I=1}^{N} \frac{PHRASESRECOGNIZED}{PHRASESPRESENTED}}{N}$$

where N = (# phrases) * (# replications). The VERBEX recognizer was given three chances to recognize a phrase.

4.6 RESULTS:

The experiment yielded interesting results with respect to the capability of the overall system. Tabular data was collected, then processed to yield graphical information. For the initial run with the first subject, recognition rates scored in the eighty to ninety percent range, with recognition rates decreasing with distance. When the experiment was run with the second subject, however, poorer results were seen. Figure 7 indicates recognition rates for both subjects with respect to the 23% vocabulary complexities. With respect to the second subject, it is possible that two training passes were not sufficient to capture the more salient characteristics of his voice. Two More additional training passes were performed with the second subject. The results indicate that the cause of the unusually low recognition rate was most likely due to insufficient training. Following the two additional training passes, recognition rates for the second subject approached that of the first subject. The training problems associated with users not familiar with voice recognition technology has been documented in the manufacturer's literature.

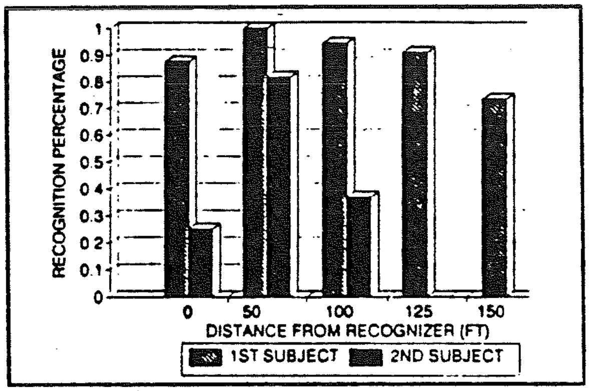


FIGURE 5 - 23% complexity results before retraining second subject.

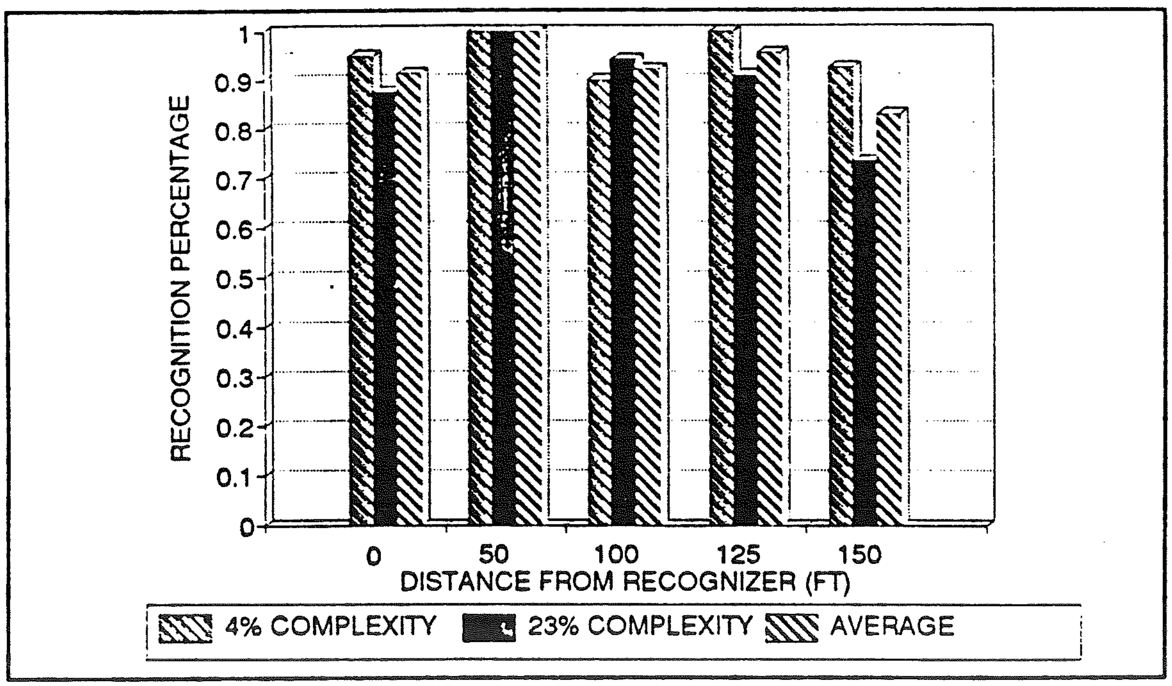


Figure 6 -- Graphical results of experiment.

The data indicates that vocabulary complexity has an effect on recognition ability, as does distance. From the data collected it may be said that:

- HIGHER COMPLEXITY YIELDS POORER RESULTS
- INCREASED DISTANCE FROM THE RECEIVER IMPEDES RECOGNITION
- OPTIMUM RECOGNITION DISTANCE UNDER EXPERIMENTAL CONDITIONS IS 50 FEET, WITH A MAXIMUM PRACTICAL DISTANCE OF 125 FEET (UNOBSTRUCTED)

5.0 DEVELOPMENT OF A MATRIX BASED PROCEDURE FOR THE DETERMINATION OF ORBITER PROCESSING TASK COMPATIBILITY WITH VOICE DATA ENTRY

5.1 INTRODUCTION

As demonstrated elsewhere in this report, Voice Data Entry (VDE) is a rapidly developing technology that has potential application in a wide variety of industrial situations. Possible applications in the orbiter processing "problem reporting" task have been illustrated in the project activities.

The bulk of the research to date in the field of VDE has been devoted to the development of equipment and software to enable VDE. Little work has been done to develop guidelines to enable an organization to systematically evaluate the tasks or processes it performs to allow an identification of those tasks that are most compatible with VDE. As pointed out by Hollingum and Cassford (1988) the basic speech recognition hardware and the software necessary to interface the speech hardware with the computer typically represents a relatively small part of the total cost, less than 30%, of the system. The remaining 70% is involved with interfacing the hardware and software with the on-going process. Careful selection of the processes is therefore necessary to ensure that VDE implementation costs do not become exorbitant.

Careful and systematic selection of tasks to be enhanced via VDE technology is important to ensure also that the benefits associated with the VDE implementation are maximized and that the impact of problems associated with the "early" or "learning curve" failures are minimized. One strategy, therefore, in selecting first tasks for VDE implementation is to pick simple non-critical tasks for first application. Another strategy, and one suggested by previous research on this project, is to introduce the new technology while simultaneously keeping the old techniques in place. Once the new technology is proven and the early failures are remedied, the older procedures can be phased out. Either of the above approaches is a useful initial approach for VDE implementation. Neither approach can be considered efficient for long-term use in the selection of tasks to be aided by VDE.

The guidelines currently available for selecting tasks are exemplified by those suggested by Hollingum and Cassford (1988) who point out that VDE applications are basically concerned with data capture. The VDE technology must be considered for application in a process in the same manner as, or as an alternative to, keyboard entry, bar-code reading, and magnetic strip reading. Factors that need to be considered in evaluating and deciding between these alternatives are typically presented in a list form such as that below:

- NEED TO HAVE EYES FREE FOR OTHER TASKS
- NEED TO HAVE HANDS FREE FOR OTHER TASKS
- NEED FOR MOBILITY, ESPECIALLY IF TASK INVOLVES MATERIAL HANDLING OR DIFFICULT ACCESS
- TASK REQUIRES DIRECT COMPUTER INPUT

- TASK REQUIRES IMMEDIATE FEEDBACK OR USE OF RESULTS FROM DATA INPUT

Utilization of a list of criteria such as that above enables a decision maker to classify a task as a viable candidate or not a candidate for enhancement by VDE. Such a list does not provide a mechanism for categorizing a large set of tasks in a operation in a manner that would enable the tasks to be ranked according to their degree of compatibility with VDE procedures.

A major activity of the VDE component of this project has been the development of a scheme that would enable a VDE compatibility index to be readily determined for the orbiter processing tasks performed at the Kennedy Space Center. The availability of this index value for each task will readily enable decisions about which activities could be enhanced by VDE, which activities are marginal candidates for VDE application, and which are unlikely candidates. Within each of these categories the index values could be used to establish a "ranking" of the tasks.

5.2 EVALUATION SCHEME FORMAT AND MACRO CRITERIA

The approach developed by the Industrial Engineering Department at UCF is a variation of the widespread matrix evaluation procedure developed by the Oregon Productivity and Technology Center of the Oregon State University (Safford, 1986). Matrix evaluation schemes have been widely used over a long period of time in the multi factor evaluation of alternatives in industrial processing situations (Reed, 1961).

Three macro criteria are suggested for initial use in the matrix procedure to enable arriving at a measure of a data entry tasks compatibility with VDE:

1. BENEFITS: Improvements to NASA's mission and Shuttle System operation expected from improvements in the information obtained from the data system. Consideration of "benefits" will limit itself to beneficial factors not normally measured in dollars (e.g. improved scheduling, task time reduction, etc.).
2. FEASIBILITY: The ease with which VDE can be incorporated into a task.
3. COST: The increase or decrease in dollar costs associated with incorporation of VDE into a data entry task.

These macro criteria could of course be considered simultaneously when evaluating a candidate data entry tasks compatibility with VDE. It is suggested, however, that each of these macro criteria be considered separately for all candidate data entry tasks. "Scores" on each macro criteria can then be systematically combined to allow an overall measure or index of compatibility to be derived.

The evaluation procedure calls for a task being considered to be assigned a score from 0-10 on each of the macro-criteria. Score sheets, described below, systematically guide the decision maker in this scoring process. After the task is scored on each of the macro-criteria a weight is assigned to each of the criteria. These weights totaling 100% are

determined by consensus from a group of "experts." The weights are then multiplied by the respective scores and the products are summed to furnish the "VDE compatibility index." The results of the aforementioned review and computation procedure would be summarized in the "VDE compatibility index matrix" form shown in figure 4.

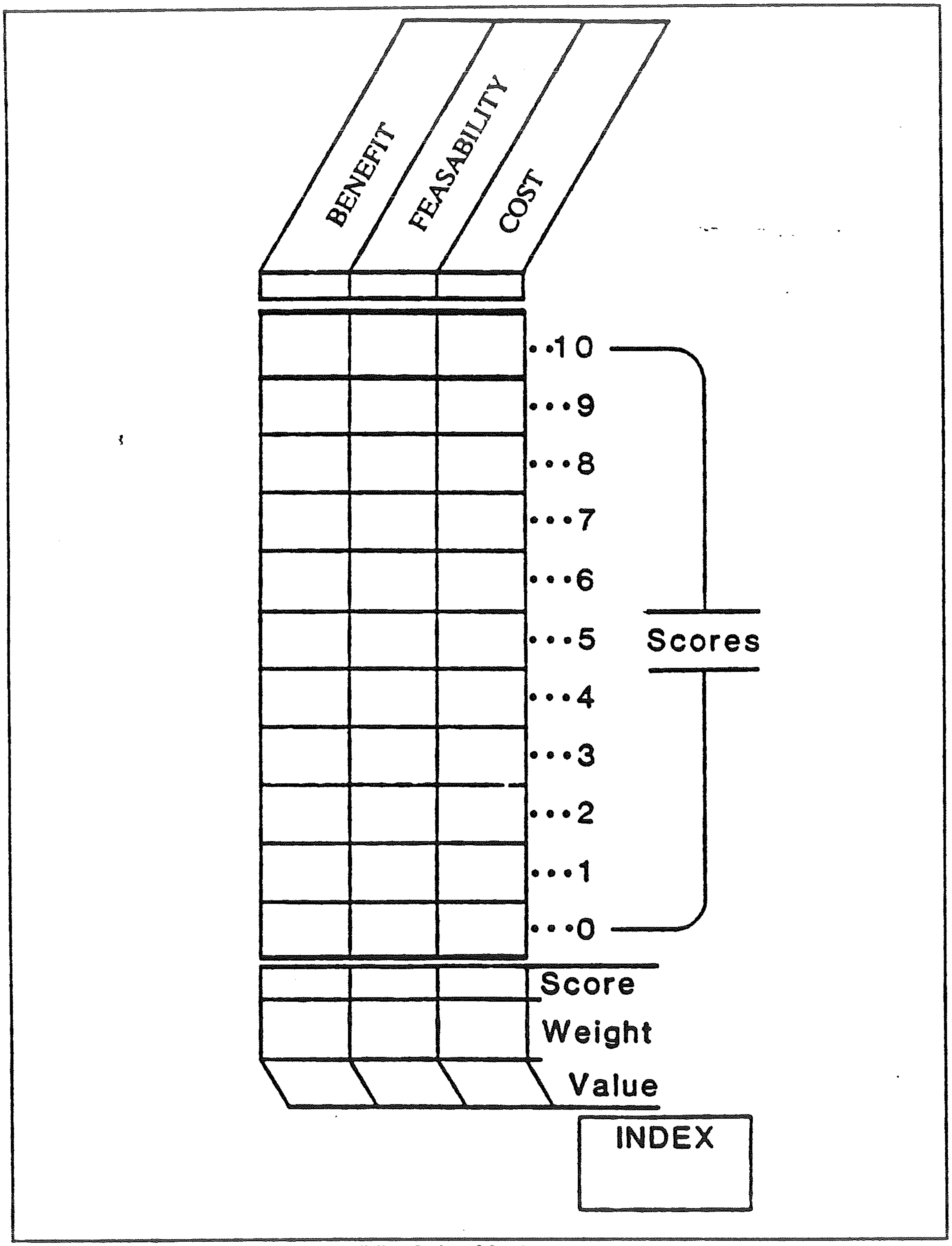


Figure 4 -- Voice Data Entry Compatibility Index Matrix.

5.3 DETERMINATION OF A "BENEFITS" SCORE

In order to utilize the matrix approach for delineating a VDE compatibility index for an orbiter processing task it will first be necessary for the evaluation team, the composition of which is described later in the report to assign a score from a scale of 0 to 10 to the benefits likely to be achieved by enhancing the task with VDE.

To enable this assignment of a score to be done in a systematic and consistent fashion it is suggested that the evaluation team consider the factors identified in table 5. These factors are expressed in the form of nine questions concerning the likelihood of experiencing potential benefits from VDE implementation.

TABLE 5 -- FACTORS TO BE CONSIDERED IN DETERMINING BENEFITS SCORE		
A.	TASK PERFORMANCE BENEFIT FACTORS	
	1)	Would the task performance be improved by VDE enabling more "free" use of hands
	2)	Would the task performance be improved by VDE enabling less eye movement away from task
	3)	Would the task performance be improved by audible feedback to operator
	4)	Would task performance time be reduced
B.	TASK QUALITY IMPROVEMENT BENEFIT FACTORS	
	1)	Would data entry errors be reduced due to VDE real time error checking
	2)	Would VDE result in reduction of transcription errors
	3)	Would "precision" of information obtained by operator be improved due to VDE
C.	OTHER BENEFITS ASSOCIATED WITH VDE AND COMPUTERIZATION	
	1)	Would VDE and its associated computerization better define task procedures for the operator
	2)	Would data be more instantly accessible and useful to other orbiter processing users
	3)	Would the direct computer access enabled by VDE be beneficial

If none of the potential benefits are likely to be achieved then a score of 0 would be appropriate. If all nine of the benefits are likely then a score of 9 would be appropriate. Six of the potential benefits would warrant a score of six. Adjustments to the initial score derived from tallying benefits would be made in the following manner. If a benefit would

not be possible in the process as it is currently structured by might be realized with minor modifications to the process fractions of a point (e.g. ½) could be assigned in the score. Adjustments could be make for unusually large significant benefit improvements by assigning more than a single point to a benefit. Similarly, disbenefits, loss of a current benefit, resulting from VDE implementation would be reflected by subtraction of points from the base score of benefits. Potential benefits not listed in the table would also be recognized with points. Adjustments to the base score would be noted by the observation team on a comments sheet and the reasons for the adjustments would be briefly explained.

After systematically considering the benefits in the aforementioned manner the evaluation team would round their score to the nearest integer on the range 0 to 10 and mark the score sheet of the task accordingly.

5.4 DETERMINATION OF A FEASIBILITY SCORE

The feasibility or ease of incorporating VDE technology into an existing orbiter processing task would be assigned an integer value in the range of 0 to 10 in the following manner which is similar to that employed in the assignation of the benefits score.

The evaluation team would first consider the feasibility factors presented in table 6. A base score of 1 times the number of factors present with respect to the task being considered would initially be assigned. For example, if five of the factors listed are present an initial score of 5 would be assigned. Additional factors contributing to feasibility identified by the team, or for potential applicability of factors given minor process modifications, would be considered in the same manner as described in the determination of a benefits score.

TABLE 6 -- FACTORS TO BE CONSIDERED IN DETERMINING FEASIBILITY SCORE	
1)	Is the task already computerized and the task process already one of data entry
2)	Does well defined vocabulary exist for the information collected in the task
3)	Is the vocabulary limited to a size that can be readily handled by available VDE equipment
4)	Is a structured vocabulary sequence in place for the information to be obtained
5)	Is there a limited "speaker" population
6)	Is the speaker population stable
7)	Is the speaker population composed of persons with "similar" speech patterns
8)	Is there readily available physical access to the computer

As with the benefit score, the feasibility score developed by the team would be rounded to the nearest integer in the range 0 to 10 and an appropriate mark for the score would be made on the score sheet.

5.5 DETERMINATION OF A COST SCORE

An integer score in the range of 0 to 10 would be determined by the evaluation team in a manner similar to that described for the benefit score and the feasibility score. The cost factors shown in table 7 will be used in the scoring procedure by the evaluation team.

TABLE 7 -- FACTORS TO BE CONSIDERED IN DETERMINING COST SCORE		
A.	EQUIPMENT COSTS	
	1)	Higher than ordinary VDE equipment costs
	2)	Higher than ordinary additional hardware or equipment costs (including additional computer equipment required by VDE)
	3)	Additional requirements for remote transmission, portability, or special computer linkages
B.	PROGRAMMING COSTS	
	1)	Extensive programming requirements for application program module
	2)	Extensive programming to handle highly variable message structure
C.	TRAINING COSTS	
	1)	Operator training costs
	2)	"Computer" training costs
D.	PROCESS DEFINITION COSTS	
	1)	Task restructuring to enable VDE
	2)	Vocabulary and structure formulation and development

5.6 EXAMPLE APPLICATION

A generic example will serve to illustrate the manner in which the observations and computations associated with the determination of the VDE compatibility index are summarized in a matrix form. A data collection activity associated with orbiter processing is examined by the review team and scores of 8, 6, and 3 are recorded respectively for the benefit, feasibility, and cost criteria. The team using a nominal group technique approach determines that weights of 45, 40, and 15 are the appropriate ones to assign to the three criteria. These values are entered in the matrix shown in figure 5. Appropriate products of weights and scores are formed and are added together to form an index in the range 0 to 1000.

More specifically in this example, the "benefit" score of 8 is multiplied by its weight of 45 to produce a product of 360. Similar procedures for the "feasibility" and "cost" columns produce products of 240 and 45 respectively. The sums of the products for the criteria produce an overall "VDE compatibility index" of 645 for this task.

The availability of this index will assist the decision maker in determining the viability of the task for actual VDE and will be useful in "ranking" this task with respect to others being considered for VDE.

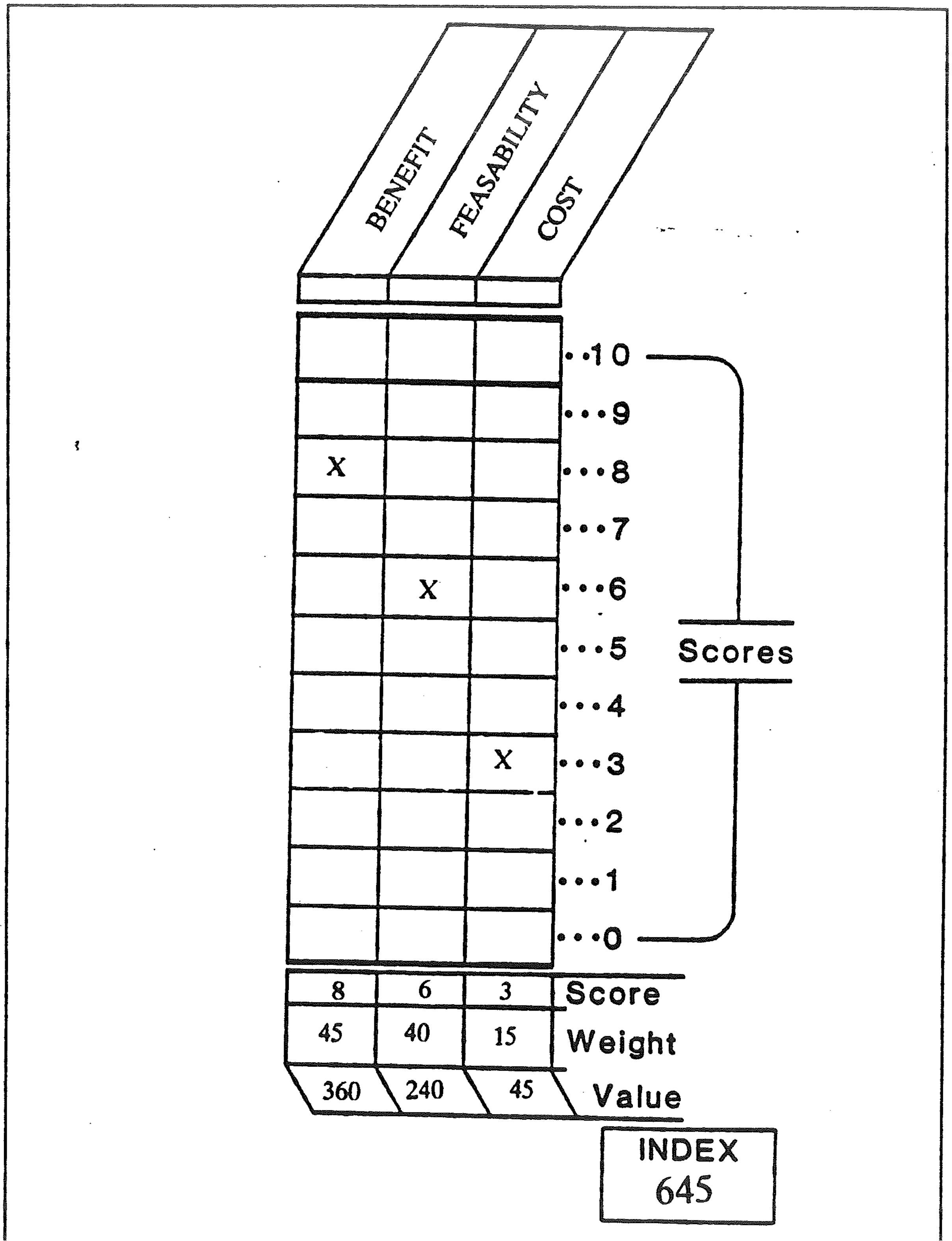


Figure 8 -- Generic example of a VDE compatibility index matrix calculation.

5.7 SUMMARY AND SUGGESTIONS FOR IMPLEMENTATION

The evaluation methodology developed in the current period of the VDE component in this project should provide a useful means for determining VDE compatibility index for orbiter processing tasks. As indicated in this report, the use of this evaluation procedure would best be undertaken by a team of individuals knowledgeable about the evaluation procedures and the orbiter processing tasks. It is suggested that the team be composed of at least three members. Possible composition of the team would include:

- A KSC representative familiar with orbiter processing procedures
- A Lockheed representative familiar with the orbiter processing procedures
- A University of Central Florida, or other representative familiar with the evaluation procedure and orbiter processing activities

The aforeméntioned team would be responsible for scoring each task with respect to VDE benefits, feasibility, and cost. The team would also be responsible for assigning weights to each of these three macro-criteria for each task that is evaluated.

The evaluation team would also have some responsibility for refining the process after a period of on-site trial usage. Such a trial would involve consideration and definition of additional macro-criteria to the matrix and refinement, revision, or addition to the tables of factors to be considered in scoring each criteria.

As mentioned previously, the availability of a systematically derived numerical VDE compatibility index should assist the decision makers responsible for determining VDE applications. The index values and procedures do not of course replace sound managerial judgement but are designed to aid the process requiring consideration of a number of quantifiable or semi-quantifiable factors.

BIBLIOGRAPHY

Banks, Peter M. and Wiskerchen, Michael J., *The SIORA Program*, Stanford University, California, 1988.

Hollingum, Jack and Cassford, Graham, *Speech Technology At Work*, New York, Springer-Verlag, 1988.

Lockheed Space Operations Company, *Thermal Protection Subsystem - Problem Reporting System*, Standard Practice Instruction #QA-002(3)K, 1990.

Reed, Ruddell Jr., *Plant Layout, Factors, Principles, and Techniques*, Homewood, Illinois, Richard D. Irwin Inc., 1961.

Safford, Robert R., Editor, *Productivity by the Objectives Matrix*, Corvalles, Oregon, Oregon Productivity and Technology Center, 1986.

VERBEX Voice Systems, *Grammar Development Manual (# 5631)*, Edison, New Jersey, 1990.

Appendix III

Cavity Digitization Report

Appendix III.1 Cavity Digitization Interim Report

Cavity Digitization

Technical Feasibility Study

Interim Report

November 1990

Cavity Digitization at KSC
Technical Feasibility Study

Interim Report

INTRODUCTION:

The objective of this study is to determine the technical feasibility of building a device which through the use of laser or optical based technology and geometric arrangement will collect 3-D descriptive information of a tile cavity in the orbiter. The information can be analyzed and translated into a machining program to cut/mill a replacement tile to fill the cavity.

This document summarizes the findings of the study and recommends a course of action for KSC and UCF.

SYSTEM ELEMENTS:

- Digitization Apparatus
- CAD/CAM
- NC milling machine

Regardless of the Digitization apparatus, elements 2. and 3. are the same, and can be acquired/ developed separately. The outcome of the digitization can be in the form of a data file which can be graphically displayed and/or supplied directly to a CAD/CAM system.

Digitization Apparatus:

A large number of devices was surveyed and studied for the digitization application at KSC. The devices based on laser triangulation were the most promising. Two methods/systems are described here which differ on their technology, arrangement of their components, digitization density (expected accuracy) and the set-up.

1. High Speed Laser Scanning (System One):

An integrated System composed of:

- Non contact laser scanner
- System controller
- Graphics Display
- Coordinate Measuring Machine or an X,Y movement mechanism

Through a synchronized scanning design, a single point laser beam is projected onto the surface of the orbiter (the cavity and its surrounding tiles). A diffuse reflection is collected back by the system and imaged onto a linear photodiode array. The position of the Gaussian peak of the scattered beam on the array provides the Z coordinates for the surface. An oscillating mirror is used to deflect the laser spot along the surface on one axis. The mirror position provides the X coordinate. The position of the scanning head is used to determine the Y coordinate for each point scanned. A CMM may provide the known Y movement. In the cavity application where digitization has to be performed overhead, a mechanism has to be devised to provide and measure the Y position.

The developer of this technology is the National Research Council of Canada (NRCC) and has given licenses for system developments to two Canadian companies (HYMARC LTD of Ottawa, Ontario and SERVO-ROBOT of Montreal, Quebec).

The technology provides for a turn-key system to be designed and to mount on a Coordinate Measuring Machine (CMM) and/or a robot arm to scan an object in a fixed position. Scanning is non-contact and will vary in the width depending on the required accuracy. Data can be collected in a step and repeat mode or continuous mode. Data collection rate is 10,000 points per second. The system may feature a real time graphics display, automatic view integration, data storage, data transfer through an Ethernet port and data formatting.

A sample cavity was digitized using the Hyscan 3-D laser Digitizer of Hymarc. The experiment produced promising results. The attached figure shows an isometric view of the cavity extracted from the generated data points. The System cost is $100,000 without the CMM.

Using such system for the cavity digitization will require developing a mounting mechanism which can be either in contact with the orbiter or used as stand alone (non contact).

This system described above will be identified as "System One" in the system evaluation section.

2. Image Processing-Based Digitization (System Two):

In this technique, the previous knowledge that the cavity is composed of planars is used. A digitizer using an image of displayed fringes, a frame grabber, and image processing software, sufficient points are digitized to produce the cavity surfaces. Through software the interactions of these surfaces are used to generate edges representing the boundaries of the cavity.

There was no system in existence that could be evaluated experimentally. Lockheed Missile and Space (LMSC) of Palo-Alto, California uses a similar technique in one of their current projects. For this reason it is assumed that LMSC is capable of building a system with these features.

This system will be refereed to as "System Two" in the system evaluation section.

APPLICATION CRITERIA AND SYSTEM EVALUATION:

1. Static Accuracy: Measurement error inherent to the digitizer.

    System One:
    Claimed accuracy by the developers are within the required accuracy of 0.005". Sample test verifies the claim. High density scanning enables capturing surface curvature. Processing and displaying surface feature is on-line.

    System Two:
    Based on the Step and Gap device, accuracy is within range assuming cavity is made up of plane surfaces. Curved surfaces can not be measured. Image Processing is done off-line.

2. Dynamic Accuracy: Measurement error due to changes between cavity and digitizer coordinates during digitization.

    System One/ Two:
    Although there was no quantification of the orbiter oscillation, the speed of the laser scanning / digitizing should reduce the dynamic error. In either of the two systems it is recommended that a stable positioning system be developed for digitization. A contact system to the orbiter is recommended whenever possible to enable stability.

3. Physical Requirements: Digitizer and positioner weight, standoff, reflectivity,compactness and their effect on efficient digitization.

    System One:
    Requires building a replacement for the CMM used in the laboratory demonstrations. An x,y,z mechanism capable of moving the scanning head in one direction (linear movement) will be required. The mechanism movements should be coordinated with the scanning procedure (equivalent to an inverted CMM). The software supplied with the system is capable of integrating multiple scans. Weight of the scanning head is manageable.

System Two:
Requires building a mechanism similar to the step and gap, however much larger to accommodate the entire cavity and to provide for enough projectors to avoid the shadow effect. A structure is required to house the system elements with fixed angles. Calibration may be required before use.

4. Ease of Use:

System One/Two:
Both systems will require user participation in the design to assure its acceptability by the technicians. Beyond the standard user friendly software methods, human factors considerations and rules for human- machine interface have to be considered in each of the development stages.

5. System Reliability:

System One:
System components are off-the-shelve and the system has been in use for a number of years in other applications. The fact that there was a digitization experiment which resulted in a CAD output through scanning proves a relatively reliable integrated system. An overall reliability for the space shuttle cavity application cannot be estimated at this stage.

System Two:
System component has to be integrated and a system has to be built. No system was tested.

6. Cost:
System One:
Rough estimate of $200,000. This will include the development of the mounting mechanism.

System Two:
Not available.

CONCLUSIONS:

1. Cavity digitization and producing a replacement tile is technically feasible using the laser triangulation technique and CAD/CAM technology.

2. Laser scanning is more accurate than sample digitization and the speed of digitization should offset the effect of orbiter oscillation/ vibration (reason sighted for other system failures).

3. Off the shelf laser scanning equipment exists, however special mechanisms must be built for the cavity application.

4. Extensive experimentation is required, on simulated test-bed using tile technicians, before commissioning the system.

RECOMMENDED COURSE OF ACTION:

1. KSC should initiate a design and development phase for cavity digitization system using the laser scanning or image processing based technology. The NRCC technology is recommended at this stage.

2. The University of Central Florida (UCF) and Lockheed Space Operation Co. (LSOC) would cooperate in the system development.

3. UCF, LSOC and NASA personnel would cooperate in software modification and on site experimentation.

JUSTIFICATION:

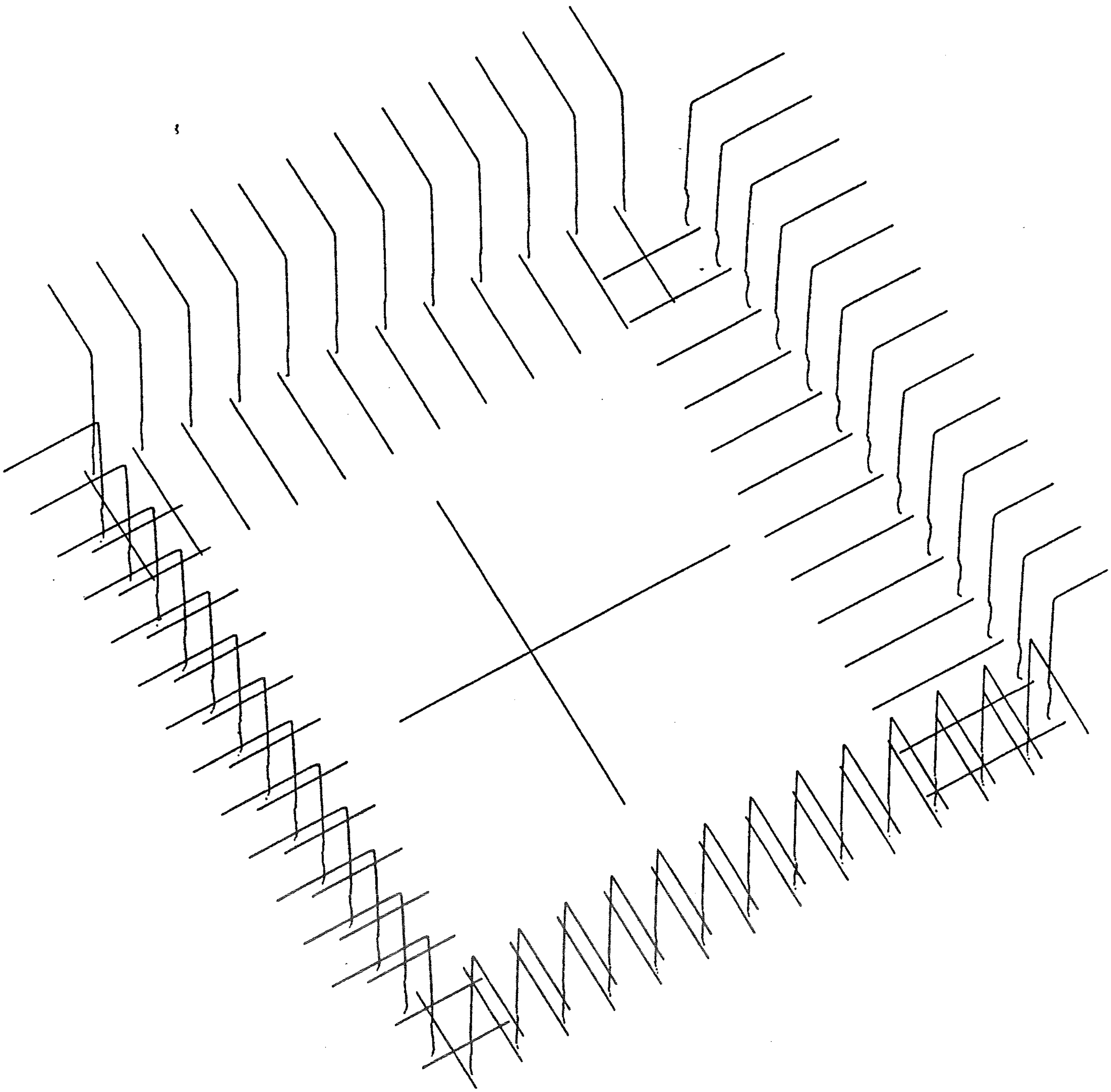
- Both UCF, LSOC, and NASA researchers are more familiar with the technology and the problems associated with its application at KSC than other potential developers.

- UCF had acquired seed funds from FHTIC for researching the problem and had developed an initial feasibility study.

- UCF maintains a laboratory testbed for any experimentation of the system.

- UCF maintains a library for equipment, specifications, studies,..etc. which should be of great value in the development phase.

- UCF is in the process of acquiring a CMM and a set of laser measuring equipment to be used for calibration and experimentation.



Lockheed Space Operations Company
1100 Lockheed Way
Titusville, Florida 32780

November 26, 1990
KKW.9043HOSNI

Dr. Yasser Hosni
Industrial Engineering Department
College of Engineering
University of Central Florida
P. O. Box 25000
Orlando, Florida 32816

Dear Dr. Hosni:

I had reviewed your summary report to NASA and the project plan setup by your group at UCF and our group here at LSOC regarding the Cavity Digitization project. By this letter I am informing you that we are fully committed to the project over the span of its duration according to the plan. We have budgeted the amount of $125,000 for the project during its first year, and we expect a similar amount in the second year. It is our understanding that UCF will participate in this project supported by the "Photomechanics in the Manufacturing and Repair of Space Shuttle Tile Protection System" study currently funded by Florida High Technology and Industry Council (FHTIC) and the productivity study supported by NASA/KSC.

We congratulate the team project on the conclusion of the feasibility phase, and we are looking forward to a successful development phase.

Sincerely,

Ken Wagy

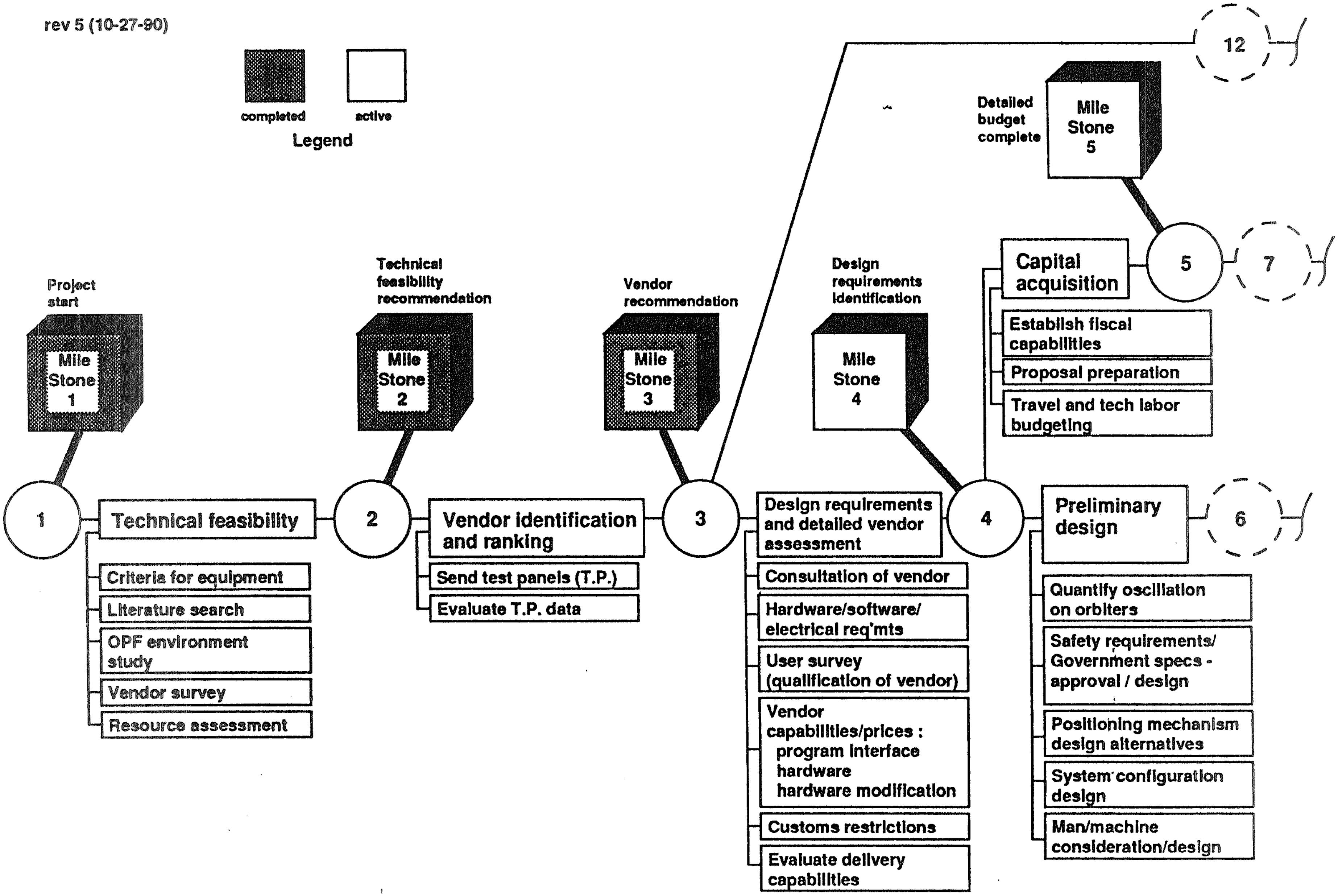
K. K. Wagy,
Director

cc: Mr. Tom Davis (PT-AST)
Mr. Tim Barth (TV-PEO-12)
Dr. Ed Hecker (TM-PCO-2)
Dr. S. Rice (UCF/COE)

Appendix III.2 UCF/LOSC Project Plan

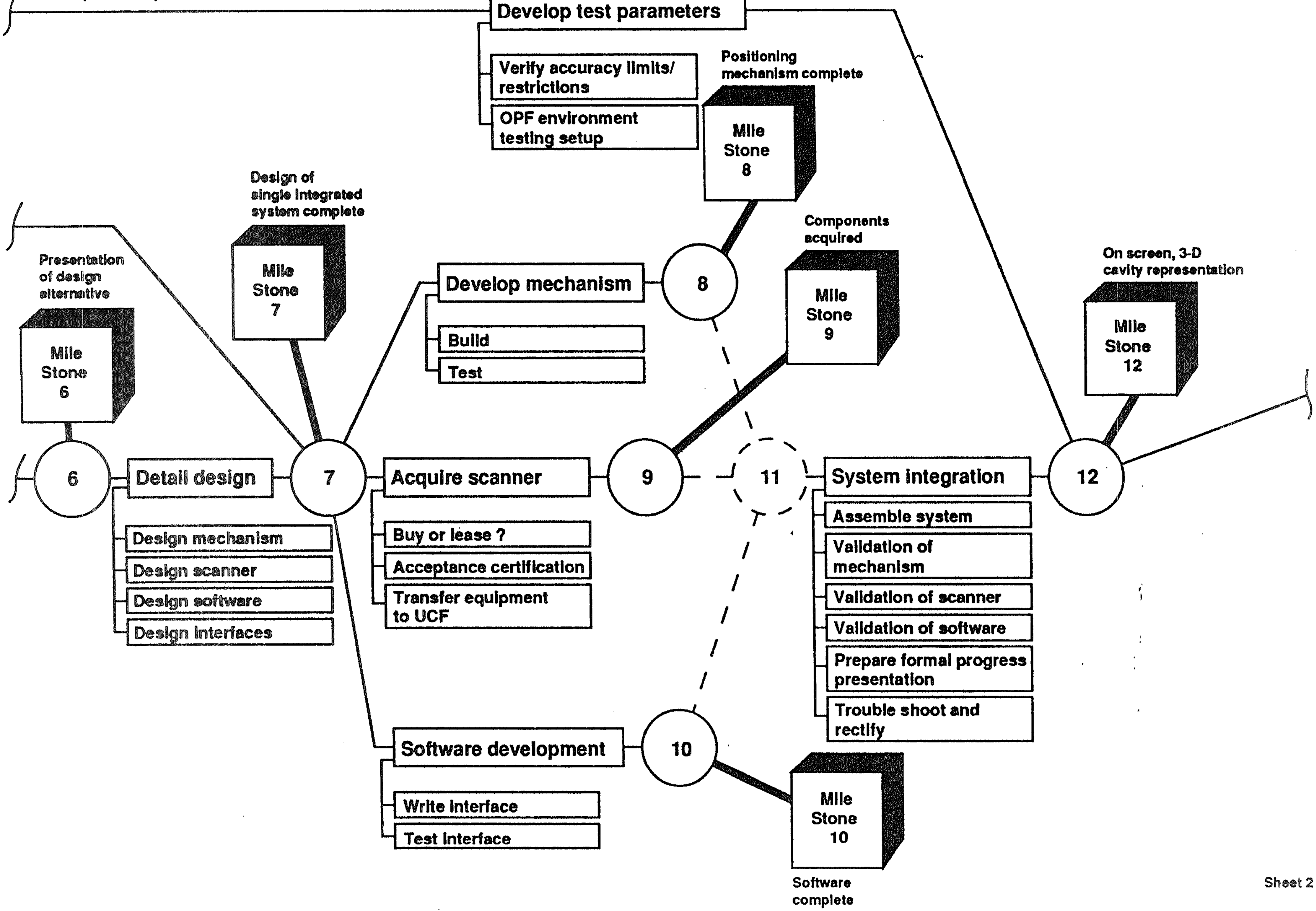
LSOC/UCF
CAVITY DIGITIZATION

rev 5 (10-27-90)



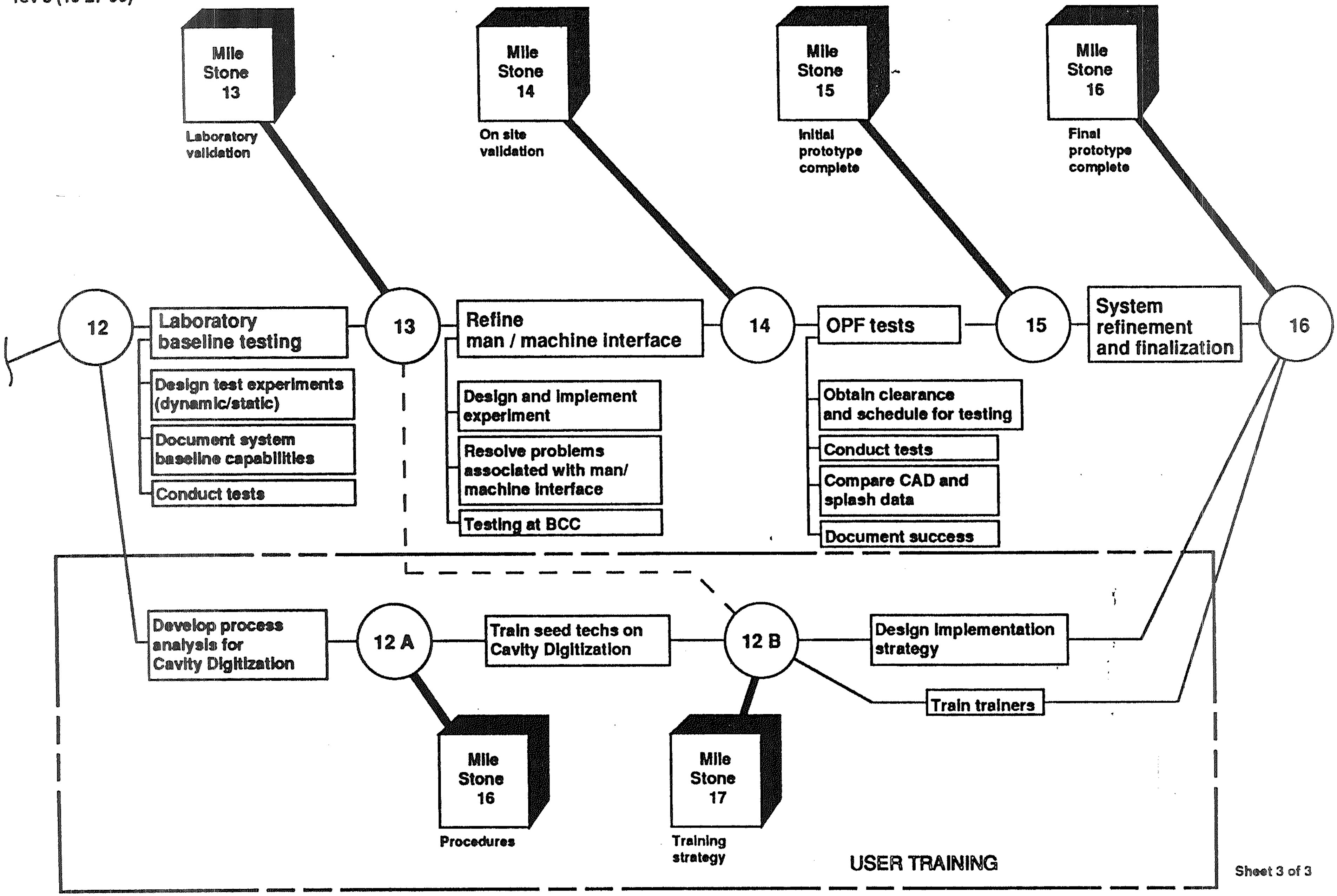
LSOC/UCF
CAVITY DIGITIZATION

rev 5 (10-27-90)



LSOC/UCF
CAVITY DIGITIZATION

rev 5 (10-27-90)



Sheet 3 of 3

Following is the description, duration and tasks associated with each activity. The layout of the description is as follows :

1. Activity name.
2. Nodes (i,j).
3. Duration.
4. Outcome (milestone).
5. Tasks, steps and procedures.
6. Responsibility.

I. ACTIVITY : TECHNICAL FEASIBILITY

NODES : (1,2)

DURATION : 12 weeks

OUTCOME :

Technical report detailing techniques, vendors, criteria for evaluation, literature search, similar experiences and recommendations.

TASKS, STEPS AND PROCEDURE :

- Study literature & previous experience.
- Conduct initial vendor survey and identify equipment.
- Study problems associated with on site application (OPF environment).
- Assess Resources available.

RESPONSIBILITY :

- UCF

II. ACTIVITY : VENDOR IDENTIFICATION AND RANKING

NODES : (2,3)

DURATION : 12 weeks

OUTCOME :

Vendor recommendation, identify all possible vendors by technique and ranking with respect to performance.

TASKS, STEPS AND PROCEDURE :

- Detailed assessment of vendors.
- Test panel preparation and send out for testing and verification.
- Assess results with respect to performance, ease, feasibility, constraints and limitations.
- Visits with selected vendors.

RESPONSIBILITY :

- UCF

III. ACTIVITY : DESIGN REQUIREMENTS AND DETAILED VENDOR ASSESSMENT

NODES : (3,4)

DURATION : 4 weeks

OUTCOME :

Design requirements and functions of system elements compatible with the highest ranking vendor equipment.

TASKS, STEPS AND PROCEDURE :

- Consultations with highest ranked vendor.
  (visits, telephonic conversations, correspondence, etc.)
- Conduct user survey and assess.
- Identify Hardware/Software/electronic requirements.
- Obtain prices, Costs, and Capabilities.
- List and resolve problem associated with imports if applicable.
- Evaluate delivery schedule capabilities.

RESPONSIBILITY :

- UCF/LSOC

IV. ACTIVITY : PRELIMINARY (CONCEPTUAL) DESIGN

NODE : (4,6)

DURATION : 8 weeks

OUTCOME :

Set of alternative design approaches. The Current trends are:

1. Hand held device, with contact or non-contact to the orbiter.
2. Stand alone device - contact of non-contact - which may be completely integrated on a special stand.

TASKS, STEPS AND PROCEDURE :

- Brainstorming sessions between LSOC, UCF, NASA, and the vendor.
- Experiment on the orbiter with the intent of obtaining quantifiable data to the orbiter oscillation/vibration.
- Set limits on design compatible with government specifications and obtain approvals (the use of laser, contacts to orbiter, schedule, etc.).
- Set limits on system performance (accuracy, device use by technicians, etc.).
- Produce a number of design approaches compatible with design specifications.

RESPONSIBILITY :

- UCF/LSOC

V. ACTIVITY : CAPITAL ACQUISITION

NODES : (4,5-7)

DURATION : 6 weeks(continuous)

OUTCOME : Secure the necessary funds and budget for the project

TASKS, STEPS AND PROCEDURE :

- Approach interested parties (LSOC, UCF).
- Prepare and submit proposals (FHTIC, NASA,others).
- Follow-up with proposals.

RESPONSIBILITY :

- UCF/LSOC

VI. ACTIVITY : DETAILED DESIGN

NODES : (6,7)

DURATION : 8 weeks

OUTCOME :

One single design will be selected, elements breakdown of the system. Elements classified as: purchased or developed. For those developed elements, detailed drawings will be produced. Software to be developed is identified by function, and its interface with other acquired/CAD software used in tile manufacturing. Final integrated assembly drawings.

TASKS, STEPS AND PROCEDURE :

- Evaluation of each of design approaches will take place through consultations between equipment vendor, LSOC, UCF, and NASA using limits of design and system performance identified in (4,6).
- The Optimal design will be selected and approved by all concerned parties. (optimality will be with respect to performance, productivity, practicality, man/machine interface, feasibility and cost).
- Identify standard elements which may be acquired as ready made.
- It is expected that the main elements of the system will be:
    - Mechanism for scanner movement and its control.
    - Scanner and its controls.
    - Software for system elements integration and control.

Interface between system and other complementary systems (CAD/CAM & technicians).

- For hardware elements to be developed by LSOC and/or UCF, detailed specifications, drawings, etc. will be produced by part, and assembly. For software elements to be developed by LSOC and/or UCF detailed specification of program function and data (input and the expected output) will be produced.
- For elements to be acquired through buying and/or leasing complete specifications and vendor identification will be produced. It is expected that the scanner vendor will be the major element subcontractor, however others may be involved.

Based on the macro identification, the system elements are:

1. Mechanism
2. Scanner
3. Software

P.S. This activity can be considered as a baseline for configuration management.

RESPONSIBILITY :

- UCF/LSOC

Three activities are identified per the network associated with the development phase.

(7,8,11) Development of the mechanism.

(7,9,11) Acquisition of the scanner as the major component for outside acquisition.

(7,10,11) Development of Software.

These development activities for system elements once completed, will be followed by a System integration activity (11,12).

Following are the details of each of these development activities (7,8-11), (7,9-11), (7,10-11) & (11,12).

VII. ACTIVITY : DEVELOPMENT OF THE MECHANISM

NODES : (7,8-11)

DURATION : 8 weeks

OUTCOME :

Complete positioning mechanism. Such mechanism may include all the necessary hardware in reliable material, motors, and measuring system. Depending on the design, the mechanism may receive its instructions for positioning the scanner holder through computer program, or would have the capability of identifying its relative position once it moves (either mechanically or automatically).

TASKS, STEPS AND PROCEDURES :

- Translation of the detailed design into an assembled mechanism. This will require the acquisition of material, machining of the parts, as well as assembling. It is expected that more than one mechanism will be built (possibly 3) at the same time. Depending on the design approach, it is possible to use a robot-type mechanism to avoid the development of new software. For such case, a gripper would have to be manufactured to fit the scanner.
- The mechanism will then be tested using local test data.

RESPONSIBILITY :

- LSOC

VIII. ACTIVITY : ACQUIRE SCANNER

NODES : (7,9-11)

DURATION : 6 weeks

OUTCOME :

Scanner compatible with the detailed design will be acquired. Depending on the cost, the capital committed and the expenditure, the scanner can be acquired through purchasing or leasing it for the duration of the project.

TASKS, STEPS AND PROCEDURE :

- Consultation with the vendors and negotiating prices, capabilities, delivery schedule, shipment schedule, tentative ownership agreement, etc.
- Transfer of the contracted components to the UCF laboratories where initial experimentation will take place.
- Experimentation, functional testing, and acceptance of the scanner using standard test beds, and familiarization with the equipment.

RESPONSIBILITY :

- UCF

IX. ACTIVITY : SOFTWARE DEVELOPMENT

NODES : (7,10-11)

DURATION : 8 weeks

OUTCOME :

Complete software package made up of a number of programs to accomplish design functions. Expected functions are in controlling the scanning head, integration of scanning pictures, data base development, CAD representation, and user interface.

TASKS, STEPS AND PROCEDURE :

- For each program:

  (a) Logic design : data flow and/or representation on the screen, and developing test data,

  (b) Coding of the programs making the maximum use of standard packages.

  (c) Testing the programs individually as well as in integration with other programs.

- Consultation with component/scanner developer to acquire the necessary design parameters during all stages of program development.

- Consultation with technicians/users for the most efficient way of user communication and man-machine interface.

- Consultation with the next phase of cavity replacement, i.e. the milling of the tile, to design an output format compatible for tile manufacturing.

RESPONSIBILITY :

- LSOC

X. ACTIVITY : SYSTEM INTEGRATION

NODES : (11,12)

DURATION : 5 weeks

OUTCOME :

Completed integrated system (mechanism, scanner, software, user).

TASKS, STEPS AND PROCEDURE :

- Assembly of the mechanism, the scanner, and the software to form the cavity digitization system. This activity will also involve "acceptance" level bench testing of the system to insure its readiness for subsequent activities. The milestone associated with the completion of this activity will be the capability of displaying a three dimension representation of test tile cavity on a computer screen.
- Test the individual cavity digitization components (i.e. mechanism, scanner, software).
- Test the ability of the system to function as a whole.
- Rectification of any system problems, on the system, as well as component level.
- Use system to scan a test tile cavity and display the results on a display terminal screen.
- Prepare a formal presentation of progress to date.

RESPONSIBILITY :

- UCF/LSOC

XI. ACTIVITY : DEVELOP TEST PARAMETERS

NODES : (3,12)

DURATION : 5 weeks

OUTCOME :

- Identify system output parameters necessary for CAD

- Fabricate test panels complete with all critical parameters, measured to the required accuracy.

TASKS, STEPS AND PROCEDURE :

- Study the critical parameter with CAD shop.
- Build a test panel (a worst case) and measure it to the highest accuracy (include an oscillating stand) using data collected from actual orbiter oscillation.
- Document the findings as a standard for comparison.

RESPONSIBILITY :

- UCF

Page intentionally left blank

probability (or level of confidence). Experiments will be conducted in the laboratory facilities at UCF.

- Perform the necessary analysis dictated by the statistical experimental design. A detailed description of system baseline capabilities and the reference conditions of the baseline capabilities test will be prepared.
- Operational analysis, time measurement, and expected gain in productivity.

RESPONSIBILITY :

- UCF

XIII. ACTIVITY : REFINE MAN-MACHINE INTERFACE

NODES : (13,14)

DURATION : 15 weeks

OUTCOME :

Laboratory prototype of the cavity digitization system complete, compatible and acceptable by the technicians who will use it, (user friendly)

TASKS, STEPS AND PROCEDURE :

- Carry on a work analysis procedure for actual technicians using the system.
- Identify and rectify any problems associated with the use of the prototype (man-machine interface). Such rectification may involve redesigning of procedure, hardware, and/or software. May require conducting the study at BCC-Space Technology Institute.

RESPONSIBILITY :

- UCF

XIV. ACTIVITY : OPF ON-SITE TESTING

NODES : (14,15)

DURATION : 17 weeks

OUTCOME :

Completion of these activities will yield an indication of on-site baseline system capabilities and will complete the initial prototype development.

TASKS, STEPS AND PROCEDURE:

- Schedule testing at the target facility.
- Obtain necessary approval.
- Conduct test.
- Compare CAD output to splash process.
- Document "lessons learned" with respect to environment, technician's acceptability, equipment performance.
- Expected number of cavity testing will be determined based on the experiment design.
- Possible application test may extend to VAB and the launch pad.

RESPONSIBILITY :

- LSOC

XV. ACTIVITY : USER TRAINING

NODES : (12,16)

DURATION : 12 weeks

OUTCOME :

- Training enough trainers and technicians (expected 5)
- A training strategy and implementation strategy
- advise for certification process.

TASKS, STEPS AND PROCEDURES :

- Develop procedure for the new device.
- Develop training strategy and train seed trainers
- Develop an implementation strategy
- Training trainers at LOSC
- Supervision of training sessions for technicians by trainers.
- Rectify and document strategies.

RESPONSIBILITY :

- UCF

XVI. ACTIVITY : SYSTEM REFINEMENT AND FINALIZATION

NODES : (15,16)

DURATION : 4 weeks

OUTCOME :

Final approved prototype with complete supporting documentation and guideline for commercialization.

TASKS, STEPS AND PROCEDURE :

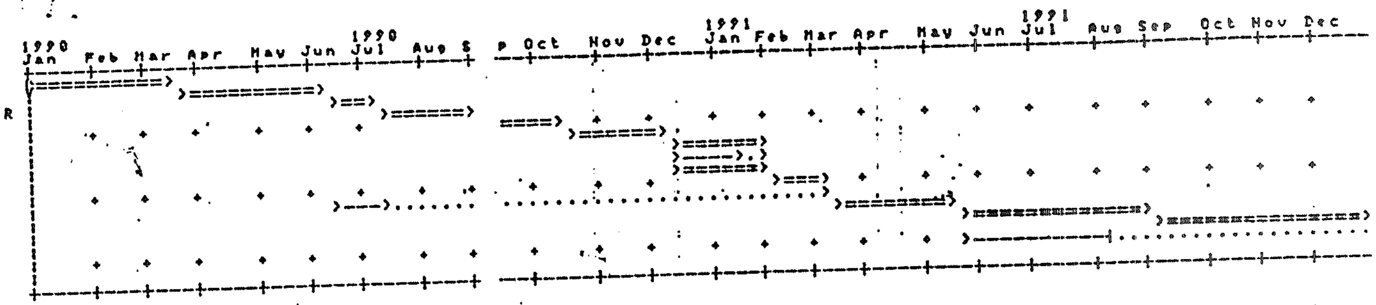
- Tasks associated with this activity will enable the final prototype of the cavity digitization system to be completed and will specify the cavity measurements and evaluation procedures necessary for implementation into the standard OMI's.

RESPONSIBILITY :

- UCF/LSOC

The details i.e. time, activity name, % completed have been used in creating the schedule. The schedule of this project has been arrived at by using Microsoft Project. The print out of the schedule is attached.

1 TECHNICAL FEASIBILITY
2 VENDOR IDENTIFICATION
3 DESIGN REQUIREMENTS AND DETAILED VENDOR
4 PRELIMINARY DESIGN
5 CAPITAL ACQUISITION
6 DETAILED DESIGN
7 DEVELOPMENT OF MECHANISM
8 ACQUIRE SCANNER
9 SOFTWARE DEVELOPMENT
10 SYSTEM INTEGRATION
11 DEVELOP TEST PARAMETERS
12 LABORATORY BASELINE TESTING
13 REFINE MAN-MACHINE INTERFACE
14 OPF ON-SITE TESTING
15 USER TRAINING
16 SYSTEM REFINEMENT AND FINALIZATION



Appendix III.3 Tally of experiments for laser equipment at UCF.

August 30, 1990

Experiment abstracts

Background: The following experiments are suggested to acquaint the members of the KSC project, cavity digitization group with the workings and capabilities of a Candid Logic Precimeter Probe.

The purpose of these experiments in the first stage is to validate the capabilities of the probe for quantifying object with 3-D information feedback. All these sub-experiments are the preliminary tests for simplifying the 3-D situation into 1-D situation. The further experiments are going to be set up based on the results in the first stage. The present experiments are intended to accomplish the following:

1. Demonstrate the abilities of the probe. To include verifying the manufactures claim to repeatability, accuracy and operating limits.

2. Establish requirements for a motion control system for laboratory manipulation of the probe.

3. Familiarize personnel with the probe, probe controller, motion controller, and the use of a personal computer to control both the probe and motion control system.

4. Demonstrate the feasibility of digitizing a cavity with the probe, and the feasibility of quantifying vibration with the probe.

5. Establish the effects of orientation between the probe and the target. Also, consider the effects of surface finish and features to probe accuracy.

The following code will be used to identify experiments. (CDE-xx-xx) The letters "CDE" stand for Cavity Digitization Experiment, and indicate this experiment has been designed, conducted or adapted by KSC project, cavity digitization group personnel. The first "xx" simply represents the numerical order in which the experiment was codified. The second "xx" is a two digit representation of the year. The first "xx" will start with "01" for each calendar year.

CDE-01-90

Title: Repeatability of the Candid Logic Precimeter Probe.

Outline: Conduct measurements of various materials, at various angles with the probe and compare the results to values obtained by a different means. Repeat experiments to determine repeatability.

CDE-02-90

Title: Optimal range determination and scanning angles.

Outline: Measure the distance to an object of ideal material for detecting the typical depth 1" or 2" over the entire range of operation. Determine the accuracy for each range by comparing the measurements to measurements of a known accuracy.

CDE-03-90

Title: Target orientation effect upon probe accuracy.

Outline: Rotate a flat target which is perpendicular to the laser beam, or tilt the probe into certain degree. Record the target, the tilt angle and indicated distance. Then make a correlation to accuracy and laser beam incidence angle.

CDE-04-90

Title: Vibration quantification in one dimension.

Outline: Vibrate a flat target with the displacement around 0.45"-0.001" and make rapid measurements with the probe by using a computer. Analyze the data to quantify the movement along the laser beam to describe the vibration.

CDE-05-90

Title: Elementary cavity digitization.

Outline: Use an X-Y table to move the probe across a rectangular cavity. Coordinate the probe measurements with a personal computer. Output the cavity representation in the form of x-y-z points.

CDE-06-90

Title: Displacement feedback

Outline: Use a triangular block with known orientation to indicate the offset of the probe from a reference point.

CDE-07-90

Title: Effect of the corner radius of a "sharp" edge.

Outline: Determine the location of an edge using the probe in conjunction with the motion control system. Then vary the radius of the corner and evaluate the ability of the probe to still detect the edge.

Conclusion

These experiments were derived from a discussion between Tom Pax, Jimmy Hwang, and Labiche Ferreira following a meeting with Dr. Hosni.

The abstracts are only a very brief description of each experiment. More work is needed to construct the experiment. Also some experiments will require specific work intensive projects to carry them out efficiently. For example, personal computer programs to interface with the probe controller and motion controller.

As with the experiments, projects should be tracked according to the following code. (CDP-xx-xx-XX) The letters "CDP" representing Cavity Digitization Project. The remainder of the code being similar to the experiment code. The last "XX" will either be not present, one letter or two or more letters. If letters are present the project would be a sub project of the original project. Example CDP-01-90-AD: This notation indicates the following projects exist: CDP-01-90, CDP-01-90-A, CDP-01-90-AA, CDP-01-90-AB, CDP-01-90-AC, and CDP-01-90-AD. The last five are all subprojects of the first one and the last four are subprojects of the second. The last four are independent of one another.

Appendix IV

Sample Publications

Page intentionally left blank

TOOL PATH GENERATION FROM SURFACE MAPPING OF AN OBJECT.

Yasser Hosni, Jueng Shing Hwang, Labiche Ferreira
Department of Industrial Engineering and Management Systems
University of Central Florida
Orlando, Florida 32816.

ABSTRACT

This paper describes an algorithm which receives scanning data of an object in the form of an X, Y, Z representation and transforms it into a tool path. A data reduction stage manipulates the data into a manageable number of points. A geometric data extraction module identifies what these points represent. These serve as an input to the surface, recognition module that identifies what kind of surface it is, i.e. planer, taper, or edge. A Numerical control (NC) program for the tool path is then generated for either the fabrication of the object or its die.

INTRODUCTION

With the advent of machine vision systems the lead time for producing the final product has been substantially reduced. A typical manufacturing cycle consists of three phases: design, production process planning and fabrication. In each of these phases, computers have widely been used.

Efforts in the past have been directed at automating the tasks right from the drawing board stage to the final production of the product. Such technologies include computer-aided design (CAD), computer-automated process planning (CAPP), and computer-aided manufacturing(CAM) including computer-aided NC programming. Among the three, CAD and NC programming are closely related to the manipulation of geometric shapes. Ideally a CAM system should be capable of accepting the geometric data in any form in order to generate NC code. However an academic survey of commercial NC programming systems revealed, that a total automation of the NC programming process does not exist.

Surface mapping or scanning is the process of digitizing geometrical information of an object's surface. A prime purpose for surface mapping of an object is to use the data for manufacturing the object or a die for its fabrication. Scanning may be done using a variety of methods such as laser triangulation or Moire method. The data generated after scanning is an X, Y, Z representation of the surface or object. The next stage is to use this data for generating a NC program for machining the object or its die.

This paper describes a method for generation of NC code from the X, Y, Z representation for a simple rectangular object through the use of an algorithm which extracts relevant information needed for the generation of NC code. The algorithm may be modified to handle a wider range of geometric entities.

The paper starts with a brief description of the surface mapping techniques followed by a description of the different modules for NC code generation.

SURFACE MAPPING TECHNIQUES

The most promising surface mapping techniques are the laser scanning (triangulation) and Moire technique.

Laser Triangulation

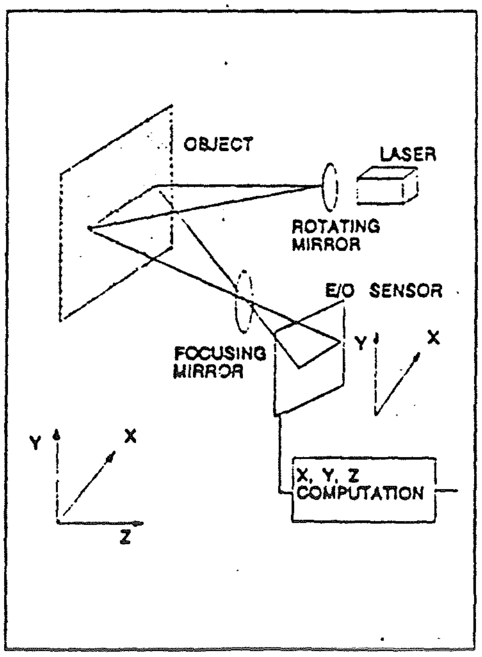


FIGURE 1 - CONFIGURATION OF LASER SCANNING DEVICE.

The basic configuration of a 3-D laser scanning device consists of a laser light source which produces a narrow light beam which is

scanned across the object to be measured through the use of a two dimensional mirror scanner. A lens collects the reflected beam and displays it on an Electro-Optical (E/O) position sensor. The linear position of the reflected light. along with the angles of deflection of the scanner are used to calculate the 3-D coordinates of a point on the object's surface. Figure 1 depicts the basic configuration of a 3-D laser scanning device.

Moire Technique

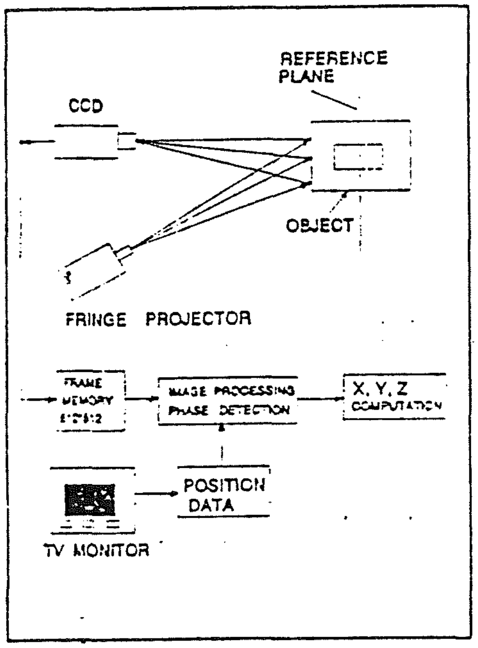


FIGURE 2 - SCHEMATIC REPRESENTATION OF THE MOIRE METHOD.

In the Moire technique, an X-Y-Z representation of an object is obtained by projecting equally spaced fringes onto the object. A 2D picture is captured by a charge couple device (CCD) and consequently digitized and stored as a frame in a computer. The phase shift of the light fringes on the image leads to the calculation of depth values for every point. Figure 2 is a schematic representation of the Moire technique.

TOOL PATH GENERATION SYSTEM

The surface mapping to tool path algorithm developed is for a simple rectangular object with planer surfaces and grooves. The object in figure 3 is used for demonstration of the algorithm. The following assumptions need to be considered:

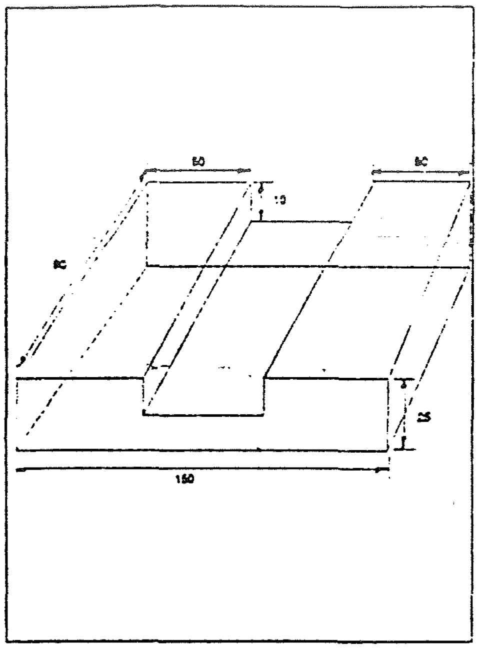


FIGURE 3 - SAMPLE PART

1) Surfaces do not have a gradient in the any direction.
2) The variation of surface depth is only in the Z direction.
3) The scanning intervals in the X and Y are predefined.
4) The number of scanning points per line is known.

The output after surface mapping an object is a X-Y-Z representation of the object. The data has to be analyzed and presented in a format which is acceptable for a CNC machine. This involves the following stages:

1) Data reduction.
2) Geometric data extraction.
3) Surface recognition.
4) NC code generation.

Both the laser triangulation and the Moire method for surface mapping result in an enormous amount of data. The data reduction stage reduces the vast amount of data generated to a manageable amount which will suffice for NC code generation. This forms the input to the geometric data extraction module. The purpose of this stage is to determine what these points represent, that is whether these points represent a line, arc, circle, or edge. The output of this stage serves as an input to the surface recognition module. This module determines the relationship between the geometric entities identified in the previous stage. In the NC code generation module the NC code is generated based on the type of the surface feature that have been identified.These stages are described in detail in the following

sections. Figure 4 depicts the overall system.

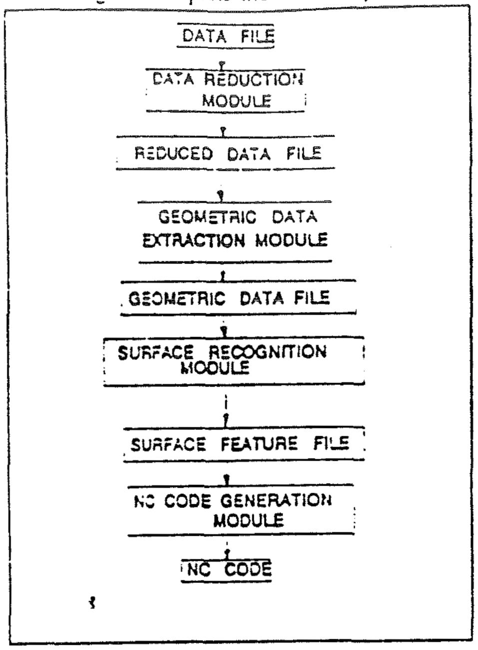


FIGURE 4 - FOUR STEP PROCEDURE FOR NC CODE GENERATION.

DATA REDUCTION MODULE

An enormous amount of data is generated after scanning. Analysis of this data revealed that the object was scanned at intervals relatively close to each other. This generated an amount of data in excess of that needed for the tool path generation. The following options are available :

1) Digitize only at critical points. Thereby limiting the points necessary for the generation of tool path statements.
2) Reducing the data to a reasonable amount. This would mean that points in the data file would be analyzed to determine if they were a part of a line, curve, etc.

This would include analysis of the X, Y, Z points contained in the data file. A description of the proposed algorithm for data reduction and stages leading to generation of NC code are described in the following sections. Figure 5 shows the flowchart for this stage. The steps involved in the data reduction algorithm include:

1) Create a record for each scanned line,
2) Consider the first point - P[i], from the data file. This point will be used as a reference point.
3) Take the second point - P[i+1].
4) Compare X[i] with X[i+1] or Y[i] with Y[i+1].
5) If X[i] = X[i+1] or Y[i] = Y[i+1] Then compare Z[i] and Z[i+1].
6) If Z[i] = Z[i+1] retain P[i]. If not go to the next point.
7) Consider P[i+1] (if P[i] has been retained), and compare it with the next point. Repeat steps 5, 6, and 7.

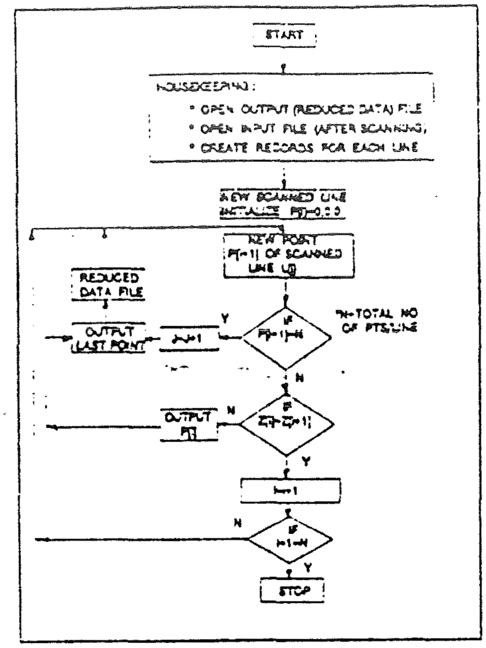


FIGURE 5 - DATA REDUCTION MODULE.

A record is created for each scanned line. The result of this data reduction stage is a reduced data file of a manageable size. This file then serves as an input for the geometric data extraction module.

GEOMETRIC DATA EXTRACTION MODULE

The next step is to determine what type of line these points represent. For example a straight line, arc, circle, etc.. For the sake of simplicity the case of a line will be considered.

The data generated after scanning a straight line consists of a string of points. However only the starting and end points are necessary to define the segment of a line in the same elevation. The line identification procedure determines if the string of points represent a straight segment of each scanned line. Figure 6 shows the line identification

procedure.

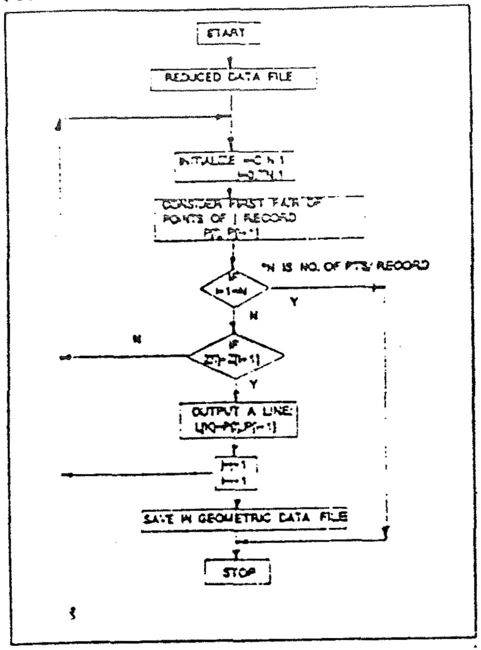


FIGURE 6 - GEOMETRIC DATA EXTRACTION MODULE.

The procedure consists of determining the start and end points of a line. The structure of the algorithm is as follows:

1) Take the first point (P[i]) in the first record.
2) Compare it with the second point (P[i+1]) of the same record.
   If
   Z[i] = Z[i+1]
   then record a line. Example -
   Line (L) =P[i], P[i+1].
3) Repeat these steps for all of the points in a record.

The output of this forms an input for the surface recognition module, which identifies the relationship between these lines.

SURFACE RECOGNITION MODULE

Three pieces of information are important for NC part programming application:

(1) the location of each surface feature,
(2) the type of each surface feature, and
(3) the relationship between each pair of surface features.

The next step after determining which geometric shape the points represent is to determine the relationship between these shapes. These lines represent an edge or are a part of a surface which might be represented by three or more points. Figure 7 shows the flowchart for this stage.

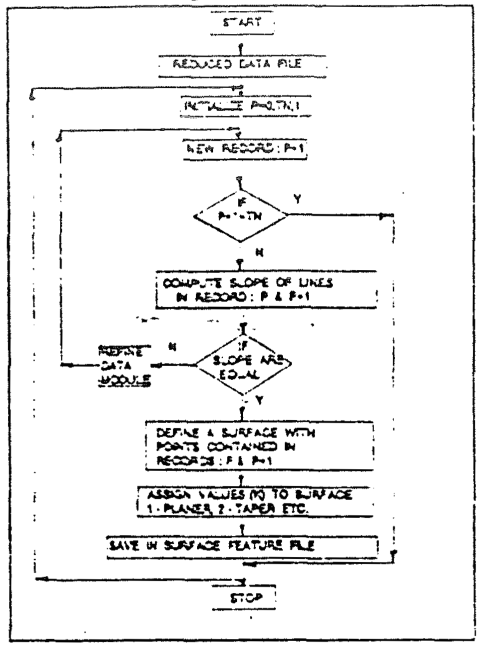


FIGURE 7 - SURFACE RECOGNITION MODULE.

The structure of the algorithm for determining the relationships between the entities is described below:

1) Consider the first set of points contained in the first record - P, (that represent the first segment of the first scanned line). Compare it with the first set of points of second record - P+1, of the second scanned line.
2) The slopes between these points are then computed. If the slopes are equal, then this could be taken to be a planer surface. If the slopes are not equal, then a change in surface orientation is recorded, i.e. taper, convex, etc.(not yet considered).
3) If the slopes of the two lines are equal then a surface is recorded which can be represented by these four points.
4) Steps 1 - 3 are repeated for each segment within every scanned line.
5) The relationship between surfaces should be stored in the order in which they are recognized. These will be called in the same order for NC code generation.
6) Each stored surface is given a code as an identification, i.e. 1 -for planer, 2 - for taper, etc.. This data forms an input for the NC part program generation algorithm which is described below. This data file is referred to as the surface feature file.

NC CODE GENERATION MODULE

Surface orientation and locations of both the start and end points of each feature are available from the surface feature file.The routine of the NC code generation algorithm is described below and the flowchart of the algorithm is shown in figure 8.

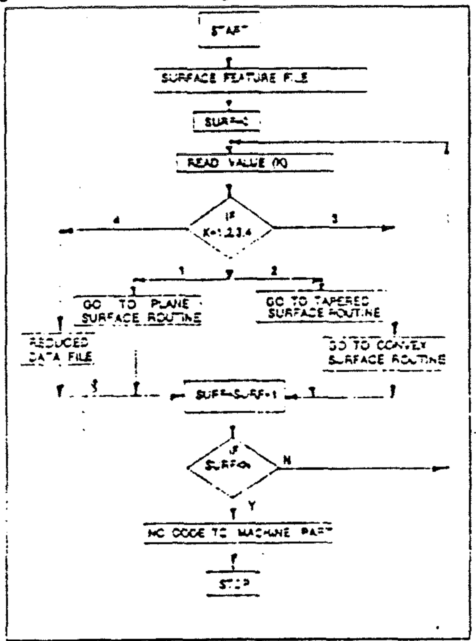


FIGURE 8 - NC CODE GENERATION MODULE.

1) Read the surface code (k) of each surface (which was identified in the previous stage).
2) Set depth of cut (d) and feed rate (f).
3) Based on the value of K, call the appropriate sub-routine for machining that surface. That is, the surface could be planer, circular, convex or concave. Depending on this value of K, the routine from the main program jumps to the sub-routine for generating NC code for the appropriate surface.
4) Repeat this procedure for all the surface features contained in the surface feature file.
5) After all the surface features have been processed, select an appropriate depth of cut for the finish cut.
6) Move the tool to the appropriate position and finish cut to generate the shape desired.

The output of this stage is a numerical control program that can be used to machine the part.

CONCLUSION

Translating data points resulting from scanning into a NC program is not an easy task. Data has to go through several algorithms/routines before its geometric configuration is identified. The algorithm presented here is an attempt to automate a process traditionally handled through user interface. The human identifies surfaces, planes, etc. and critical dimensions.

The algorithm described in this paper is for planer surfaces, where a minimum of three points are needed to define a plane. These planes are then used to define boundaries and edges.The algorithm is in its initial stage. The expanded version to handle a variety of configurations is being studied. Complex cases where surfaces with varying curvatures require previous knowledge about the surface characteristics, so that the proper algorithm will control the tool motion. A major benefit of the surface mapping to tool path algorithm is its elimination of human labor in generating NC code.

REFERENCES

[1] **Robert Mills**, 15 Ways to Link CAD and CAM, *Computer-Aided Engineering*. (Oct 1989).

[2] **Optiz, H et al.** Calculation of technological data for the computer assisted programming of NC machine tools. *Numerical control programming languages*, North Holland(1970), pp 243-267.

[3] **B. K. Choi, M. M. Barash and D. C. Anderson**, Automatic Recognition of Machined Surfaces from a 3D Solid Model, *Computer Aided Design*. 16(1), pp 81-86, 1984.

[4] **B. Chen**, ROMAPT: A New Link between CAD and CAM, *Computer Aided Design*. 14(5), pp 261-266, 1982.

[5] **David Hergert, Robert Speckert**, Coverting A Graphics Screen To a CNC File, *PROCIEM '89*, pp 99-101.

[6] **Waite Group**, *Microsoft C Programming for the IBM*, 1987.

[7] **Chao-Hwa Chang, Mickel A. Melkanoff**, *NC Machine Programming & Software Design*, Prentice Hall (1989).

ACKNOWLEDGEMENT

The authors wish to acknowledge The Florida High Technology and Industrial Council for funding the Photomechanics study from which this paper resulted.

Page intentionally left blank

Ergonomics of Hybrid Automated Systems II
W. Karwowski and M. Rahimi (Editors)

DESIGN OF A WORKSTAND FOR OVERHEAD OPERATIONS: CASE OF THE SPACE SHUTTLE HEAT-TILE REPLACEMENT

Y. Hosni[1], C. Lee[1], J. Creech[1], T. Barth[2], and C. Hill[2]

[1]Department of Industrial Engineering and Management Systems, University of Central Florida, P.O. Box, 25000, Orlando, FL 32816
[2]NASA Vehicle Engineering, Kennedy Space Center, Florida 32899

SUMMARY

Operations conducted on the space shuttle at Kennedy Space Center (KSC) involve tasks such as orbiter heat tile inspection, repair, and replacement. Although tile and other thermal protection material are used on the entire orbiter exterior, work involving the lower surfaces is the most challenging to the tile technicians since a great deal of work is done "out of position" i.e., upside down.

Access to the lower surface tile is achieved from floor level by using various designs and sizes of workstands, some portable, some not. Most often, one or two technicians are involved in a given task.

Presented in this paper is a design of a seat to be installed on a workstand for overhead tasks. The design considers ergonomic factors dealing with upper body support for the technician when performing the task in less than optimal position. Other factors are practical in nature such as portability, access to full circumference of the work platform, and height adjustability to permit optimal positioning of workstand for access to all tile on the lower surfaces for the orbiter.

INTRODUCTION

The NASA/KSC, and the University of Central Florida (UCF) are presently investigating improvements in the design of workstands used in performing tasks related to the repair and maintenance of the Thermal Protection System (TPS) on the Space Shuttle Orbiter vehicles. The overall objective of the study is to expedite repairs and to improve the accuracy with which these repairs are accomplished.

TPS maintenance is performed after each flight on all external surfaces of each of the orbiters, as well as in the payload bays. However, since most damage to the TPS is sustained on the lower and flight control surfaces, particularly in the aft section, the research team is restricting its study to maintenance tasks performed from stands positioned on the shop floor.

The configuration of the orbiter and the tasks performed on it (study limited to lower and flight control surfaces) require that virtually all processes be performed in non-standard work positions, that is, overhead.

The literature searched indicates that there is little data available on task performance in these postures. In fact, the postures and positions assumed are not recognized as anthropometrically acceptable for precision work when considering the limits of human reach and vision. It is evident that technician performance may be degraded by problems associated with visibility of the work area and fatigue brought on both by working in less than optimal postures, and in elevating the eyes and rotating the neck backward to view the work area. Further, the work positions assumed by technicians, and the mobility/adjustability of the stands have significant safety implications.

WORK ENVIRONMENT

Processing of the TPS of the orbiters is unique in the aerospace industry due to several conditions existing in the Orbiter Processing Facility (OPF) environment, the design of the orbiters themselves, and the nature of the tasks accomplished. Following are some characteristics of the work environment which may affect workstand design.

1. The thermal tile are easily damaged by contact with equipment, tooling, workstands, or the skin of the technicians.

2. Many maintenance activities, both on the vehicle and the tile, are performed simultaneously; hence, the work area is typically crowded by personnel, equipment, and workstands of various designs. All of these obstacles must be "worked around." Figure 1. shows the working place around the orbiter, ready for maintenance operations.

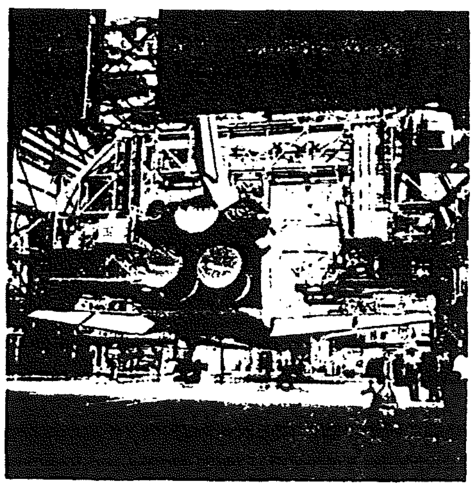


Figure 1. Work area in the OPF

3. The environment is restrictive in the materials and designs that can be used in the construction of the workstands. Primary restrictions concern the flame spread rate and out-gasing under fire conditions of the materials used in construction, as well as the mobility of the stands and the level of cleanliness that must be maintained.

4. Repair or replacement of TPS components requires extreme precision in most cases, dictating that the stands be rigid and stable, when work is performed.

5. Most tasks are presently performed in out-of-position postures, that is overhead. New stand designs should minimize fatigue complications in these postures.

6. The maintenance processes vary significantly in the surface area that is covered in accomplishing them. Repair of individual tile and removal and/or bonding of new tile is typically restricted to an area of one square foot. Measuring of the step and gap between several tile may typically have the technician traversing areas of several feet.

7. Most of the processes accomplished in TPS maintenance require the labor of only one person. However, all processes are inspected and approved by quality control and engineering personnel at numerous stages. Hence, ingress and egress from the working/inspection position must be accomplished easily and quickly.

8. Task accomplishment is heavily dominated by documentation and referral to written procedures--there must be immediate and easy access to paperwork and manuals.

DESIGN CRITERIA

1. The workstands must provide easy access to the lower surfaces of the orbiters which range from 8 to 12 feet from the shop floor in the OPF. Male and female technicians of normal height ranges use the stands.

2. The workstands should have infinite height adjustment to accommodate the range of height of the work areas, as well as the range of height and reach of the technicians. The stands' height should be easily adjustable by one person, and preferably would be adjustable from the technician's working position.

3. The design should provide postural support for the technicians in accomplishing overhead processes thereby minimizing the effects of fatigue and visibility of the work area.

4. It is recommended that the use of electrical and/or hydraulic power use be minimized for safety reasons, i.e. cables, hoses, etc., should not be extended around the orbiter on the floor of the work place. Manually powered devices is a recommended substitute.

5. If wheels are to be used on the workstands it should be of large enough diameter to allow easy passage over floor gratings used to cover equipment chases and drainage channels.

6. Proof testing - stands should be able to withstand static loadings of 375 lbs. for each person that they will accommodate without failure or significant deformation. Handrails should be able to withstand loadings of 375 lbs. applied downward at a 45 degree angle from vertical Steps and/or ladder rungs should be able to withstand static loading of 375 lbs.

7. The design of the workstand must conform to NASA standards for Space Shuttle Ground Support Equipment General Design Requirements.

8. Materials used in construction must conform to NASA standards for materials permissible for use in the OPF.

9. Human engineering criteria should conform to the military standards for Military Systems, Equipment and Facilities.

WORKSTAND DESIGN

The design strategy for the modified or new workstand is directed at increasing technician safety, decreasing the time to perform TPS processing tasks, and increasing the precision with which these tasks are accomplished.

All current and any added alternatives have, or should have, four elements included in their design:

1. The use of larger and wider wheels - minimum of 8" diameter and 2.5" width, to permit easy rolling over obstacles and zero penetration of the floor gratings.

2. Improvements in the range and ease of height adjustments.

3. Improved access to equipment and paperwork.

4. Features to allow the technician to work in a semi-reclined seating position, or as an alternative, provision for upper body support when standing while leaning backward.

The first three elements are essential. The fourth, reclining the technician, may not prove beneficial for tasks performed that traverse large areas, but should be considered essential for tasks limited to small areas such as repair of one or a few tiles, or cavity preparation and/or tile bonding to the surface of the orbiter.

Due to the space limitation in this paper, we are reporting on the design of "seat to be added to an existing workstand which is currently being used at KSC. Figure 2.a is the current workstand and the suggested location for the seat assembly. The design makes use of the unused "volume" under the stand platform in housing the seat assembly. In situations where there is a need for an operation requiring positioning of operator for long periods of time performing an overhead operation, the platform would be opened and the seat assembly would be raised to the platform level when it is to secured. Figure 2.b. shows the elements of the seat assembly, when a height adjustable mechanism (powered manually or mechanically), can raise a seated worker to the proper position underneath the orbiter. The seat is equipped with a rotating subassembly table to allow coverage of an area and it is in the proper position. The operator can then lock the table in a certain position.

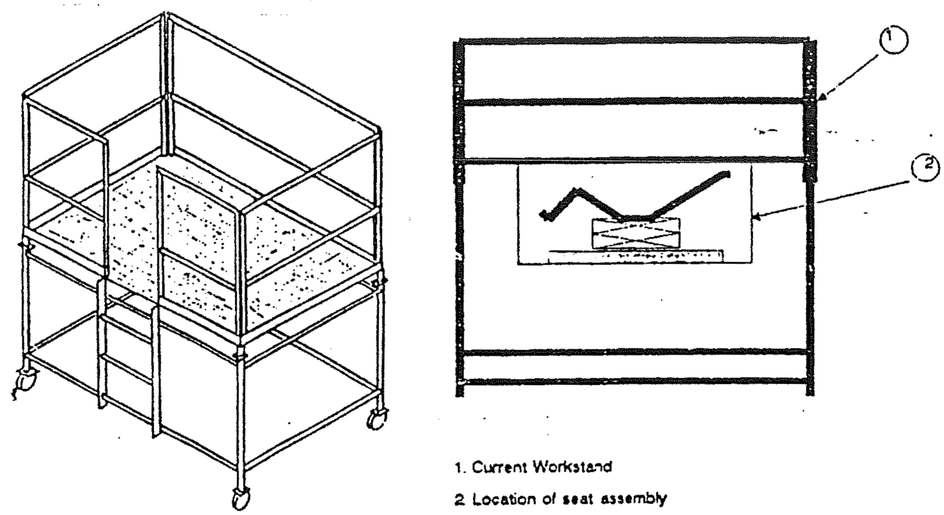


Figure 2.a.
Current Workstand and suggested location for the seat assembly.

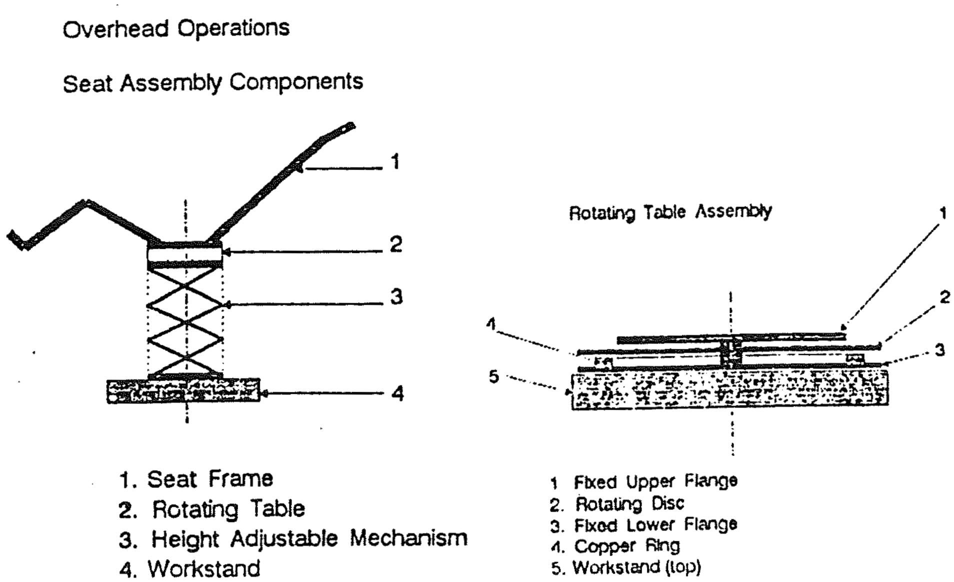


Figure 2.b.
Elements of the seat subassembly

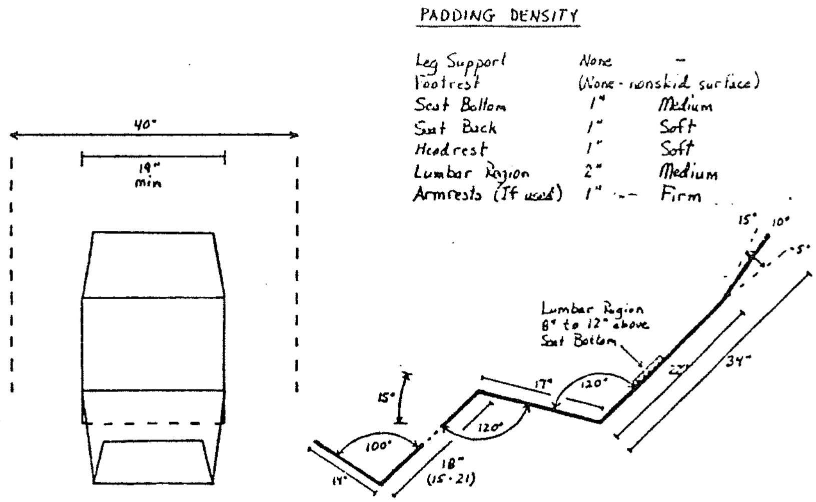


Figure 2.c.
Seat Dimensions.

Figure 2.c. is detailed drawing of the seat dimensions. It is a compromise of several criteria, but generally follow the guidelines for the average adult (lengths, angles) so as to accommodate the range of sizes for the technician population (Human scale, IE Handbook, MIL-STD-1472). Modifications may be made, but the following criteria should be adhered to:

1. With backrest angles in this range (120 degrees shown), a headrest must be used, and headrest height should not be less than 31".

2. Footrest angles should not allow knee flection of less than 100 degrees (110-120 optimum).

3. Seat bottom inclination should not be less than 15 degrees (shown) to prevent sliding downward. Angles greater than 15 degrees, if used, will make ingress or egress more difficult.

4. Footrest angle should not be less than 90 degrees (100 degrees shown).

5. Lumbar support should be maintained, via thicker padding or contouring of the backrest.

6. Knee to ankle distance should be adjustable over the range of 15" to 21".

7. Optimum headrest angle for reclined seating positions when viewing objects near the horizontal place is 10 degrees (shown). However, for this application where the technician is looking overhead, the headrest angle should be adjustable.

8. Seat width should not be less than 19" and hip clearance not less than 22".

9. Work area width should be unobstructed for a width of 40" to allow full arm movement.

CONCLUSION

The ergonomic design of a workstand should be based not only on anthropometric data but also specific requirements of the work itself and workers behavioral pattern. The space shuttle heat tile replacement represent an unique working environment with specific requirements.

This paper presented a workstand design based on the specific design criteria established for the TPS repair and maintenance tasks. As demonstrated in this paper, the general ergonomic guidelines for workstand design should be modified under practical conditions in order to improve its effectiveness. The present design can be adapted for other aerospace industries which have common problem with overhead operation.

REFERENCES

1. Diffrient, N., A. Tilley, and D. Harman "Humanscale" the MIT Press. MIT, Cambridge, MA, 1981.
2. Grandjean, E. "Fitting the Task to the Man, A Textbook of Occupational Ergonomics", 4th Ed. 1988, Taylor & Francis.
3. MIL-STD-1472C, Human Engineering Design Criteria for Military Systems, Equipment and Facilities, Department of Defense, Washington, D.C., May 1981.
4. Yasser A. Hosni, and William W. Swart; "Productivity Improvement Techniques"; NASA-KSC/UCF Cooperative Agreement for Technology Development, June 1989.
5. Yasser A. Hosni, John Creech, "Two person Workstand Design", Interim Report, UCF, March 1990.

DESIGN AND EVALUATION OF A WORKSEAT FOR OVERHEAD TASKS

C.H. Lee[1], Y.A. Hosni[1], L.L. Guthrie[1], T. Barth[2] and C. Hill[2]

[1]Department of Industrial Engineering and Management Systems,
University of Central Florida, Orlando, FL 32816

[2]NASA Vehicle Engineering,
Kennedy Space Center, Cape Canaveral, FL 32899

Workers in some industries such as the aerospace industry are frequently required to work in awkward overhead positions resulting in physiological and biomechanical stress to the workers as well as degraded worker performance. General ergonomic guidelines can be used to provide the workers with more comfortable work positions which will improve the productivity and safety of the workers. This paper presents the design and evaluation of an ergonomically designed workseat which considers ergonomic factors dealing with whole body support for the workers when performing tasks in a less than optimal position. The workseat allows the workers to perform their tasks in semi-reclined positions instead of the normal standing overhead position. A prototype of this workseat was produced and evaluated both objectively and subjectively.

INTRODUCTION

The Thermal Protection System Repair and Maintenance Procedures for the Space Shuttle Orbiters performed on the lower surface dictates less than optimal working positions for tile technicians. The technicians are forced to work in uncomfortable, fatiguing overhead positions with no back or neck support. An ergonomically designed workseat providing more body support and comfortable working positions was designed and evaluated. The evaluation of the workseat was conducted by simulating some overhead operations performed by tile technicians at NASA (KSC). Five tile technicians from NASA (KSC) participated as human sujects during the evaluation process. Their task performance and general and body part discomfort with and without the support of the workseat were measured and compared.

METHOD

Design of Workseat

From the existing ergonomics database and literatures, the following design criteria for the workseat was established (Diffrient et al., 1981, Grandjean, 1990, MIL-STD-1472C).

1. With backrests angles in approiaximately 120 degrees, a headrest must be used, and headrest height of at least 31".
2. Footrest angles should not allow knee flexion of less than 100 degrees (110-120 optimum).
3. Seat bottom inclination should not be less than 15 degrees to prevent sliding and ensure ease of egress and ingress.
4. Footrest angle should be at least 90 degrees.
5. Back support should be maintained, via thicker padding or contouring of the backrest.
6. Knee to ankle distance should be adjustable over the range of 15" to 21".
7. Adjustable headrest angle with 10 degrees where the workers are looking overhead.
8. Seat width should be no less than 19" and hip clearance of at least 22".
9. Unobstructed work area width of 40" to allow full arm movement.

Figure 1 is a detailed drawing of the seat dimensions. It is a compromise of several criteria, but generally follow the guidelines for the average adult (lengths, angles) so as to accommodate the range of sizes for the worker population. A prototype of the workseat was fabricated and evaluated.

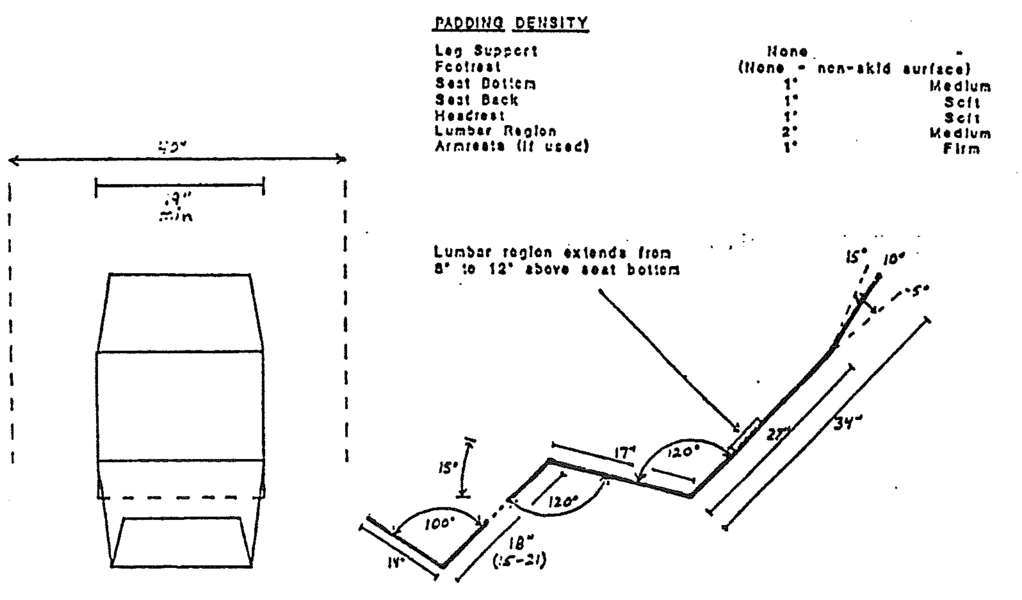


Figure 1 Workseat Design

Experimental Evaluation

An experiment was conducted through the simulation of actual Thermal Protection Repair and Maintenance Procedures performed at the NASA (KSC) Orbiter Processing Facility in addition to general application tasks. Thus, the evaluation results will be a direct reflection of the possible improvement of worker performance and comfort of NASA workers and other applicable industries.

Processes P-601, Installation of Pillow (Alumina) Gap Fillers, and P-301, Tile Installation were chosen for simulation. The two processes were chosen because of the high frequency of their occurrence and ease of simulation. In addition, Rotary Pursuit Tracking and Weighted Overhead Work tasks were also included to simulate general overhead task applications.

The entire processes of gap filler and tile installation were not to be completed during this experiment. Several of the prepatory and prefit tasks are unnecessary, irrelevant, and far too time-consuming for the experimental purposes. The work performances of the subjects was monitored throughout the processes. The type and the time duration for each task were observed and recorded using work sampling methods at thirty second intervals.

Rotary Pursuit Tracking is designed to allow for human performance measurements in the area of accuracy by simulating fine motor skills and is a commonly accepted and recognized method of measurement in the ergonomics field. The tracking device consists of a fluorescent light source rotating in a circular motion and a stylus wand with a photocell at the tip. The subjects pursued the moving light target with the stylus, attempting to keep the stylus and light moving together on the circular path. The photocell initiated the digital stop clock which measures the cumulative time of successful pursuit. By keeping the duration of the task, sensitivity of the sensor, and speed of rotation constant for all subjects, an effective measurement of accuracy was easily obtained. The pursuit tracking task was limited to two five minute sessions with a rest period of five minutes between each session, the sensitivity set at level 10, and the speed was set at 40 revolutions per minute. The amount of time the subject could successfully pursue the light source was recorded as time-on-target and was recorded at one minute intervals during both sessions of 5 minute duration.

The Weighted Overhead Task was also included as a general application experiment, in order to measure the endurance time The task readily served as simulation of overhead work involved heavy objects. The subjects were instructed to align a weighted box (approximately 15 pounds) in the cavity positioned overhead. The subject were further instructed to hold the weighted box overhead in that position as long as they possibly could without moving the box from the cavity. Contact sensors mounted inside the cavity were to signal by the light, connected in series with the contacts, when full contact was first achieved between the box and the sensors within the cavity and when that contact first faltered. This task served to measure overhead endurance time. By monitoring the time needed for alignment and the maximum time a subject could support the weighted box, an accurate measure of endurance was obtained.

Performance Parameters

The performance parameters measured were task completion time, task accuracy, endurance time, and body discomfort. The general and body part discomfort of each tile technician was subjectively evaluated upon the completion of each process.

During the experiment the performance and discomfort of the subjects during actual process activity were measured in an effort to contrast the level of performance and discomfort associated with both the typical standing overhead working position and the semi-reclined and supported position offered by the ergonomic seat.

Five volunteer tile technicians from KSC participated as human subjects. They were fully instructed as to the nature and purpose of the experiment and encouraged to perform the processes at their usual level of effort and rate.

The experiment involved five subjects and four processes; two tile repair and maintenance, P-601 and P-301, simulations and two general application tasks, Rotary Pursuit Tracking and Weighted Overhead Tasks. The whole experiment was replicated once in order to reduce the experiment error.The subjects performed the four processes in two working positions; standing and seated. The experiment required four separate visits from each subject for a time duration of approximately 1.5-2 hours each. Each visit was virtually the same except for the position in which the processes were completed. Each subject completed all four processes on each visit. During the performance of each process, the worker performance and discomfort of each subject were measured and monitored. The parameters measured were task completion time, endurance time, task accuracy, and body discomfort.

RESULTS

The Statistical Analysis System (SAS 1984) was used to analyze the data obtained during the experiments. The independent variable used was working position, i.e. standing, seated. The dependent variables were task completion time, endurance time, task accuracy, and body discomfort.

Figure 2 shows the task completion time during the tile processing. As the figure reveals, there was only an insignificant difference in the task completion time. The endurance time measured during the weighted overhead work task is depicted in Figure 3. The figure illustrates a statistically significant increase of 62.7% in endurance time. The Rotary Pursuit Task was used to measure accuracy. Figure 4 shows the accuracy percentage levels associated with both postures for both sessions of tracking. There was a significant increase of 4.2% in the amount of time-on-target from the standing to the seated position during the first session and 3.44% increase in the amount of time-on-target from the standing to the seated position in the second session. Utilizing the scaled discomfort index of range of 0 to 10, Figure 5 illustrates the average level of discomfort associated with each body part throughout the stages of the experiment. Figure 6 shows the varying levels of discomfort associated with each position by task. As illustrated by the figures, the seated position was much more comfortable than the standard standing position.

The free response comments expressed by the subject divulged a list of shared concerns. In summary, the subjects maintained the need for work and storage space in the form of trays or tables and a waste container. They also voiced concern over dripping RTV and other chemicals and falling debris. Many of technicians stated their worry of dropping tools and materials from the workseat as well as falling from the seat themselves. Arm rests, permanently tethered tools or safety net, and a chemical and debris shield could be implemented to improve the seat design.

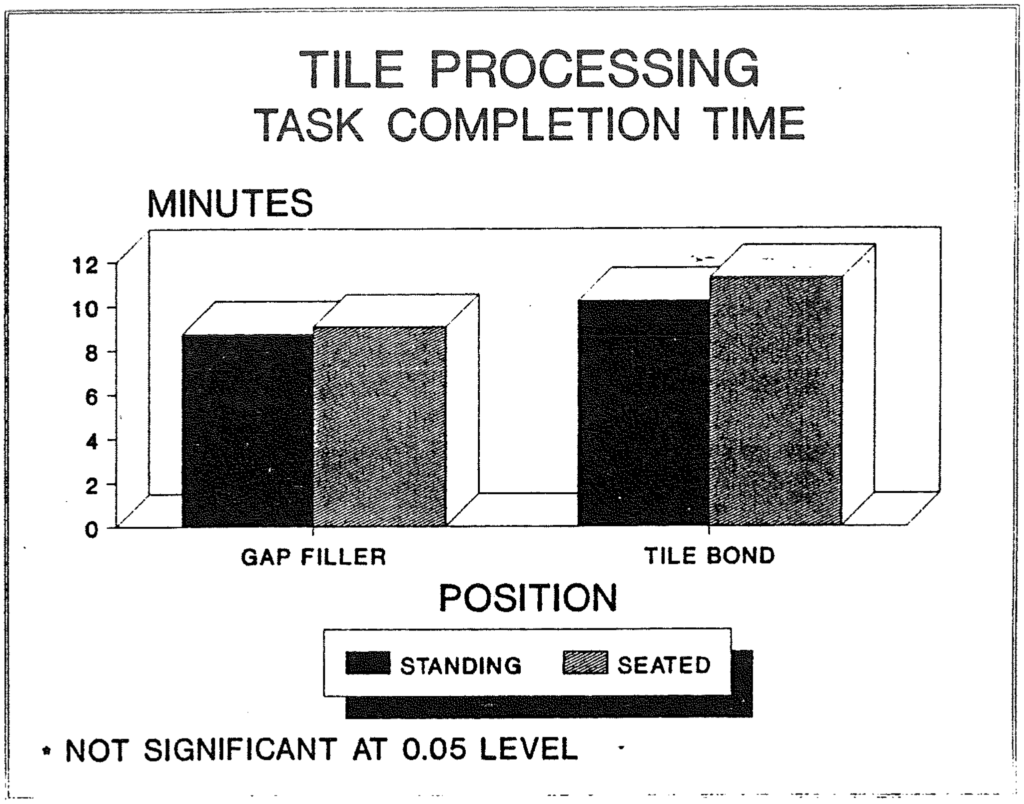


Figure 2 Task Completion Time

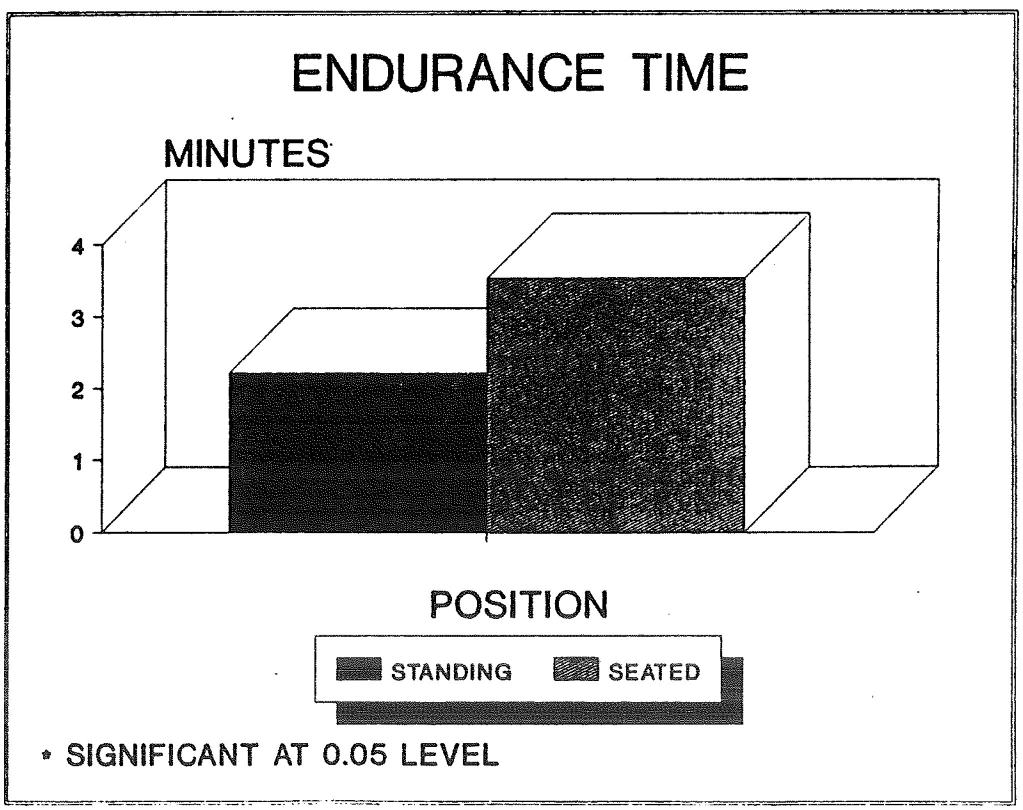


Figure 3 Endurance Time

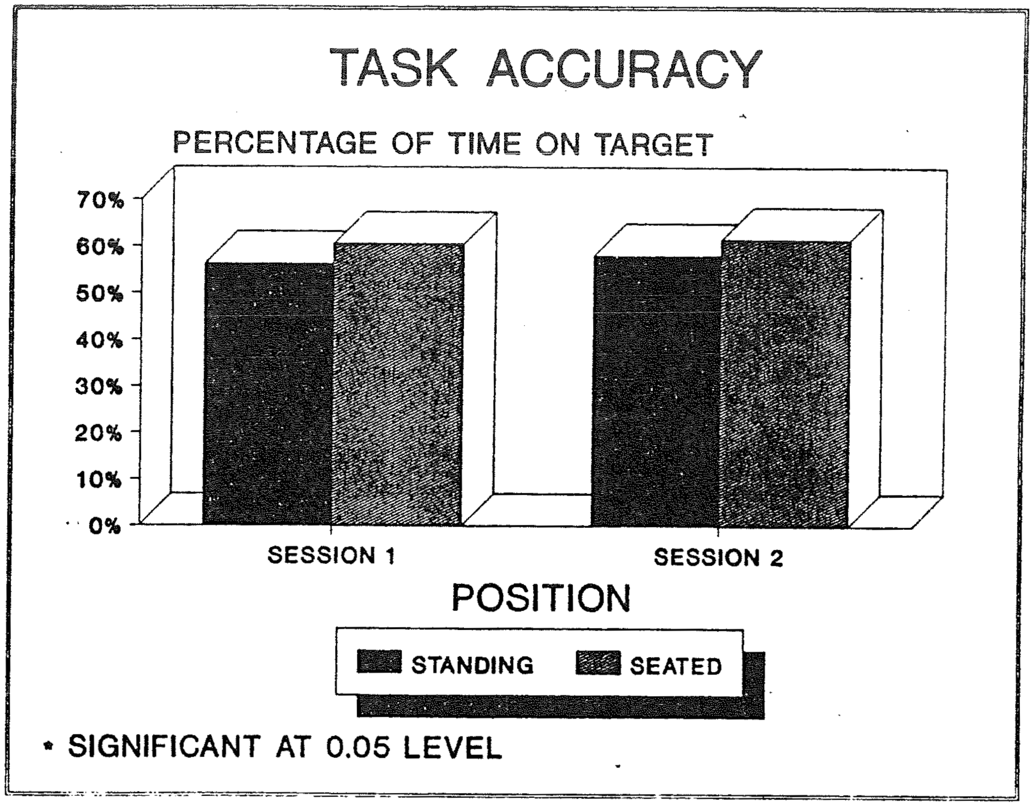


Figure 4 Task Accuracy

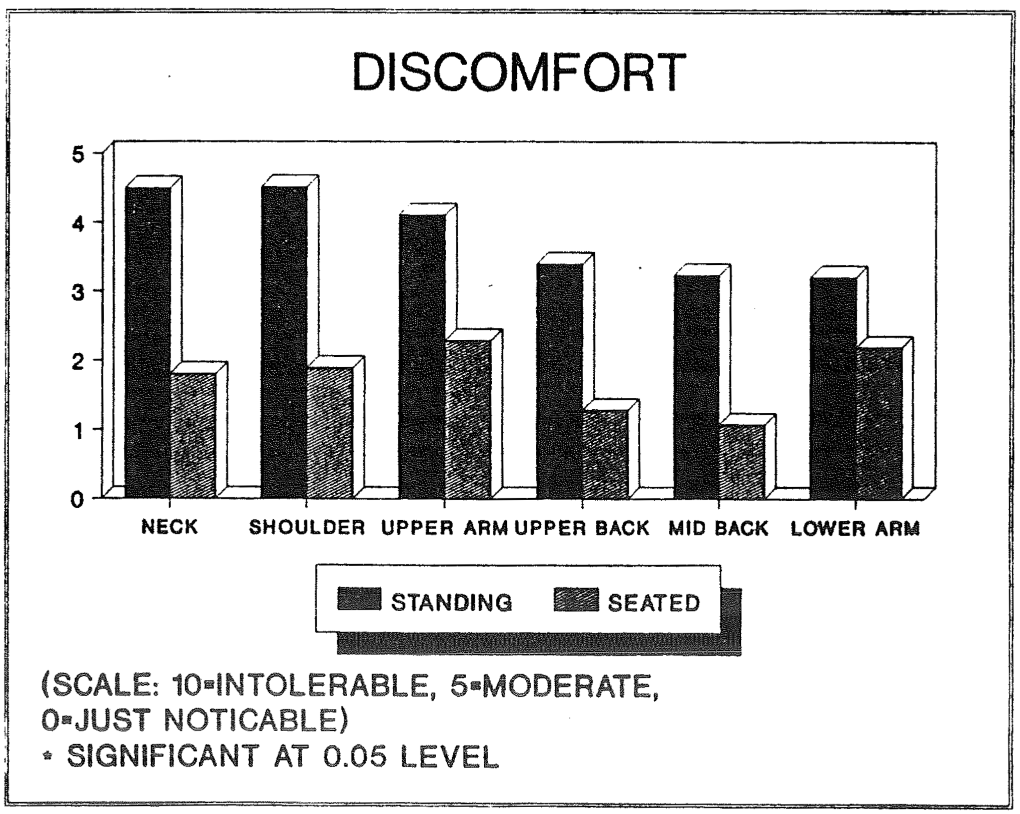


Figure 5A Body Part Discomfort

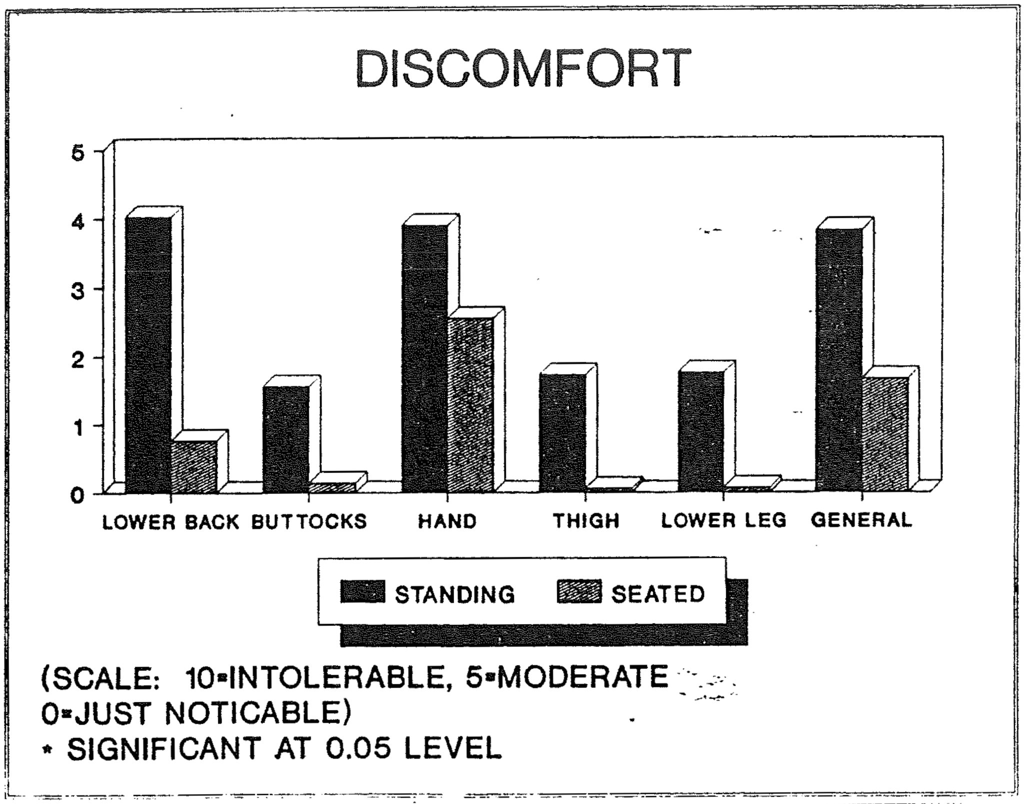


Figure 5B Body Part Discomfort

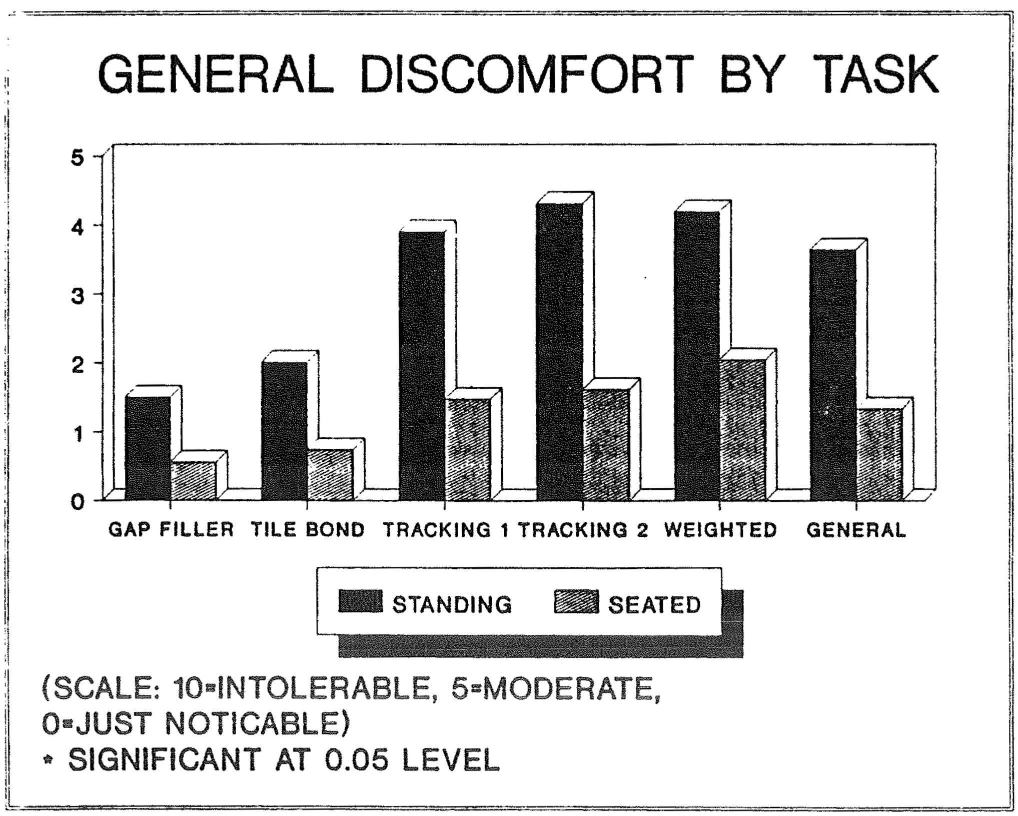


Figure 6 General Discomfort by Task

CONCLUSIONS

The seated position was significantly superior to the standard standing working position in terms of performance and body discomfort. Initially, the results of the tile processing time were not very different. However, when coupled with the large differences in endurance time and the difference in accuracy levels which were measured over a very short period of time, it is obvious that the differences would be largely magnified over the course of an entire work shift. Furthermore, the much higher level of discomfort associated with the standard working position alone points to the seated position as a better alternative.

The results of the evaluation showed that an ergonomically designed workseat improves the quality of work, reduce task completion time and necessary rest periods, and increase comfort during overhead operations.

REFERENCES

Diffrient, N., A. Tilley, and D. Harmon "Humanscale" the MIT Press. MIT, Cambridge, MA, 1981.

Grandjean, E. "Fitting the Task to the Man, A Textbook of Occupational Ergonomics," 4th Ed. 1988, Taylor & Francis.

MIL-STD-1724C, Human Engineering Design Criteria for Military Systems, Equipment and Facilities, Department of Defense, Washington, D.C., May 1981.

SAS Institute, Inc. "SAS Users Guide," Statistics. 1984 Ed. SAS Institute, Inc. Cary, N.C. 1984.

56-26

N91-70704

p-20

KSC-NASA/ UCF COOPERATIVE AGREEMENT

MATERIALS PERFORMANCE
RESEARCH PROGRAM

END OF THE FIRST YEAR STATUS REPORT
January-December,1990

Submitted by: Dr. V. H. Desai, P.I.
Mr. Kirk Scammon
Mr. Edward Principe
Ms. Lisa Sperling

Mechanical & Aerospace Engineering Department
College of Engineering
University of Central Florida

EXECUTIVE SUMMARY:

Nearly 40% of the materials related failures at Kennedy space center are corrosion related. Half of these are estimated to be stress corrosion cracking (SCC) failures. Prevalence of acidic chloride environment and a broad range of service conditions make the situation severe and complex for variety of materials that are expected to perform well.

During the first year of this research program, nine Fe and Ni based alloys were rated for SCC susceptibility by slow strain rate testing (SSR). Progress was made in determine fundamental SCC parameters such as crack propagation rate, activation energy, incubation or initiation time and threshold stress intensity through the use of innovative techniques such as lead pulsing, laser holography, and interferometry. These studies will lead to service life prediction. Also, hydrogen embrittlement susceptibility study of highly corrosion resistant Ni based alloys has initiated. The alloys being tested are candidate materials for hydrogen storage tanks. An electrochemical system for hydrogen permeation study has been setup and tested while embrittlement studies through SSR technique have initiated.

Two additional projects were identified for the second year of research. These are characterization of acidified sea water used to simulate KSC environment and study of concrete corrosion and prevention.

BACKGROUND:

Corrosion accounts for about 40% of the materials related failures encountered at NASA's Kennedy Space Center. Nearly half such failures are due to stress corrosion cracking (SCC), making it the most significant cause of material failure. SCC is followed by pitting, hydrogen embrittlement, uniform corrosion, galvanic corrosion, and exfoliation in its order of severity. This is graphically represented in figures 1 and 2. The variety of service conditions and metallic alloys employed make the situation complex requiring a concentrated effort to find failure mechanisms and provide solutions through greater understanding. Vicinity of sea water and presence of HCl acid mist following SRB blast make the corrosion conditions very severe. For example, the hydrogen storage tanks undergo almost round the year maintenance. This has lead to the suggestion of using highly corrosion resistant Ni base alloys as materials of construction for these tanks. Table I summarizes the environment handled and the variety of metallic alloys used at KSC.

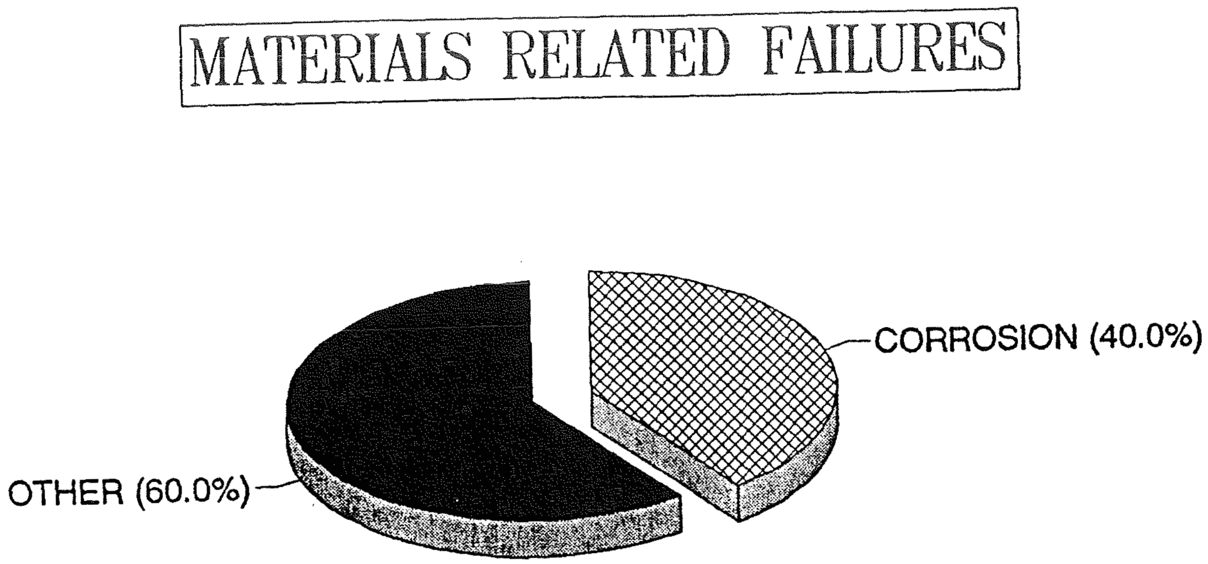


Figure 1

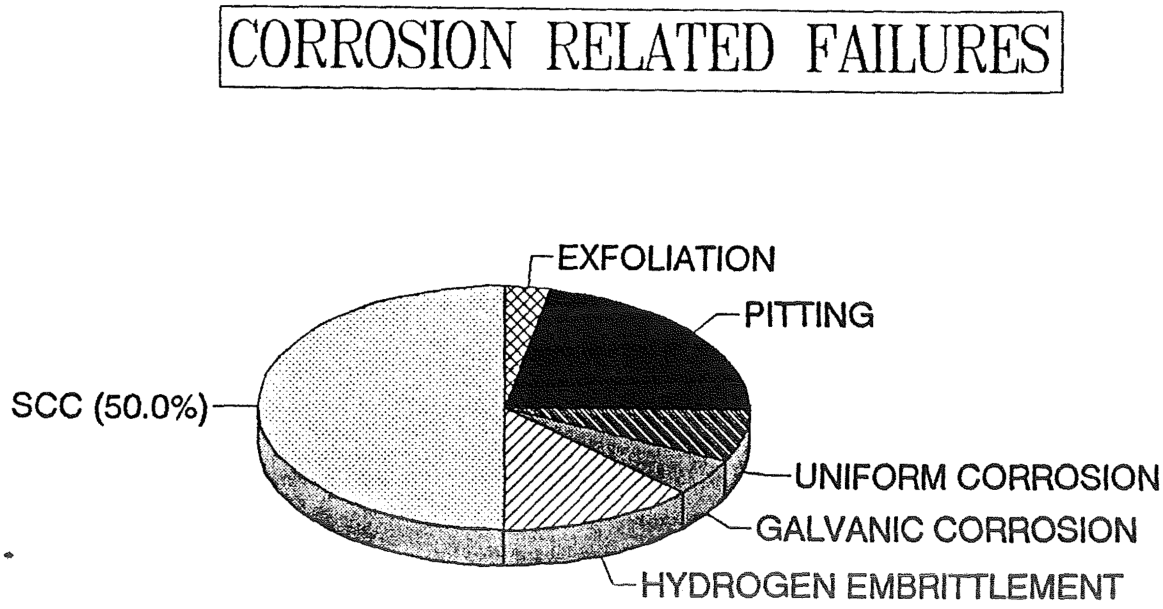


Figure 2

Complexity of KSC Situation

Environment Handled	Metallic Alloy Used
Marine	1,2,3,4,5,6
Marine + HCl acid mist	1,2,3,4,5
Gaseous Nitrogen	1,2,3,4
Gaseous Helium	1,2,3,4
Liquid and Gaseous Oxygen	1,4
Liquid Hydrogen	1,4
Gaseous Hydrogen	1,2,4
Liquid Nitrogen	1,4
Monomethyl Hydrazine (MMH)	1
Hydrazine (N_2H_4)	1
Nitrogen Tetroxide (N_2O_4)	1

Key: Metallic Alloys

1	Stainless Steels
2	C Steels and Alloy Steels
3	Quenched and Tempered Steels
4	High Nickel Alloys
5	Aluminum Alloys
6	Titanium Alloys

Table I

OBJECTIVES:

Task I

Determine the SCC susceptibility of various Fe and Ni base alloys by slow strain rate (SSR) tests in simulated laboratory environment (sea water acidified to pH 1) and rate these alloys in order of their performance for use at or in the vicinity of launch complex.

Task II

SCC studies through innovative techniques

(A) Using load pulsing method, determine the velocity of crack propagation and activation energy for both trans and intergranular mode of cracking of austenitic stainless steels in chloride environments. Also, determine the effects of temperature, environment and metallurgical variables on SCC susceptibility and relate to service life predictions and protection.

(B) Using coherent optical techniques such as laser holography and speckle photography determine the time to initiate SCC, threshold stress intensity and crack propagation rate. Relate this data to service life predictions and develop the technique to quickly assess the effect of metallurgical variables on SCC parameters.

Task III

Hydrogen susceptibility studies

Determine the susceptibility to hydrogen damage of highly corrosion resistant alloys such as Hastelloys C22, C276, G30, Inconel 625, and Inconel 718. And recommend alloys resistance to hydrogen storage tanks (3,000 to 6,000 psi) which can give at least ten years of trouble free service.

Task I SCC SUSCEPTIBILITY STUDIES BY SSR TESTS

Approach

1. Flat plate specimens of various alloys were pulled in air and acidified sea water at strain rate of ~ 10-6 per second.

2. The susceptibility to SCC was evaluated by comparing the air and sea water tests. Six criteria were used to determine susceptibility namely time to failure, maximum stress, strain at maximum stress, maximum strain, % reduction in area and fracture area ratio (FAR).

3. In order to assess the effect of strain rate, three different strain rates were utilized for studying the susceptibility of three alloys.

Progress

1. Nine alloys have been assessed for SCC susceptibility. The results of these tests are summarized in Table II. Each data point is an average of three tests and indicates % decrease in the value of respective parameter compared with the test in air.

2. Based on the six parameters listed earlier, the alloys are rated as follows:
   Highly susceptible to SCC
   - 303 steel (annealed)
   - 17-4 PH steel (H 1025)
   - 17-4 PH steel (H 900)

   Moderately susceptible to SCC
   - 316 L steel (annealed)
   - 304 steel (annealed)
   - 17-4 PH steel (annealed)
   - 17-4 PH steel (H 1150)

   Non-susceptible to SCC
   - Inconel 718 (annealed)
   - Inconel 718 (40 HRC)

3. The three alloys tested to assess the effect of strain rate indicated a great dependance of susceptibility on the strain rate chosen to run SSR tests.

ALLOY	TIME TO FAILURE	MAXIMUM STRESS	STRAIN AT MAXSTRESS	MAXIMUM STRAIN	REDUCTION IN AREA
303 ANNEALED	-71%	-34%	-75%	-68%	-75%
304L ANNEALED	NO CHANGE	NO CHANGE	-4%	-6%	-16%
316L ANNEALED	NO CHANGE	NO CHANGE	-12%	-12%	NO CHANGE
INCONEL 718	NO CHANGE	NO CHANGE	NO CHANGE	NO CHANGE	NO CHANGE
IN 718 40 HRC	NO CHANGE	NO CHANGE	NO CHANGE	NO CHANGE	NO CHANGE
17-4PH ANNEALED	-10%	NO CHANGE	NO CHANGE	NO CHANGE	-17%
17-4PH H1150	-5%	NO CHANGE	-9%	-9%	-8%
17-4PH H1025	-42%	NO CHANGE	-40%	-60%	-67%
17-4PH H900	-30%	NO CHANGE	NO CHANGE	-40%	-27%

TABLE # 2 DIFFERENCE BETWEEN AIR AND SOLUTION TESTS

4. We recommend the following practice in SCC investigation through SSR tests.

Recommended Practice for SCC Investigation by SSR Tests

1. Perform SSR experiments on each of the 50 different alloys at approx. 10^{-6} per sec. strain rate in acidified (pH 1) sea water.

2. Discard alloys that are highly susceptible as determined by a combination of ultimate strength, time to failure, reduction in area, % elongation, strain to fracture and strain at maximum stress.

3. Run additional SSR experiments on moderately susceptible and non-susceptible alloys at strain rates of 10^{-5} and 10^{-7} per sec.

4. Alloys that do not show susceptibility at any of the three strain rates are not susceptible to SCC.

5. Subject all other alloys to SEM fractographic investigation, determine FAR and rank them in order using all the available data.

Task IIA SCC STUDIES THROUGH LOAD PULSING METHOD

Approach

1. Cylindrical specimens of 316 stainless steel were subjected to periodic impulse load of about 5% over the base load which was typically maintained at about 80% yield strength of the alloy. This load pulsing was carried out while the specimens were subjected to a 5% by weight LiCl environment.

2. Tests are being conducted in the range of 110° to 170°C in a autoclave so that different test temperatures could be achieved and maintained while keeping the chloride concentration constant.

3. The crack tip blunting obtained on the fractured surface due to pulsed loading was observed under scanning electron microscope (SEM) as parallel markings. A micrograph indicating such marking in trans and intergranular areas of fracture is shown in figure 3.

4. The average distance between markings when divided by the time interval between pulsing events yields crack propagation rate.

5. A plot of velocity of crack propagation versus test temperature (Arrhenius plot) gives the activation energy required for the propagation of crack.

Progress

1. Tests have been conducted on 316 stainless steel specimens in LiCl environment at 110° and 170° in an autoclave.

2. Problems encountered in "swagelok" fittings and leaking from blow-off valve of the autoclave have been corrected.

3. SEM observations have been carried out on the fractured surface and crack propagation rates for TSCC and ISCC modes of cracking have been deduced. Table III summarizes these results. Each data is an average of several observations.

4. A preliminary Arrhenius type of plot is shown in figure 4 for TSCC. It is too early to calculate activation energy and predict the mechanism of SCC. More data at several other temperatures is awaited.

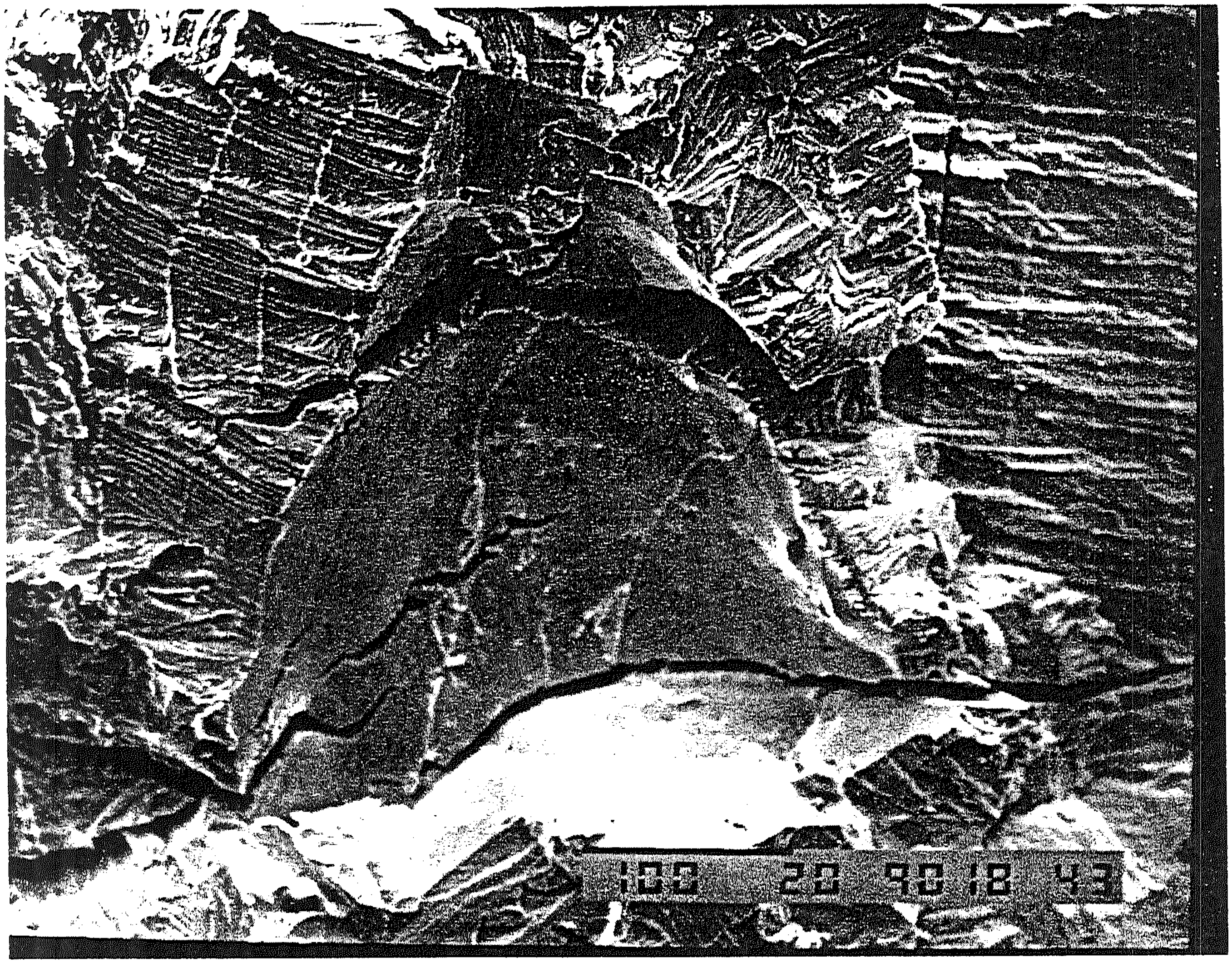


Figure 3 Crack front marking on trans and intergranular fracture

ISCC Velocity = 2 to 3 x (TSCC Velocity)

TEMP.	VELOCITY (10^-7 m/s)	
	TSCC	ISCC
170 C	2.117	4.423
130 C	1.390	4.550

Table III

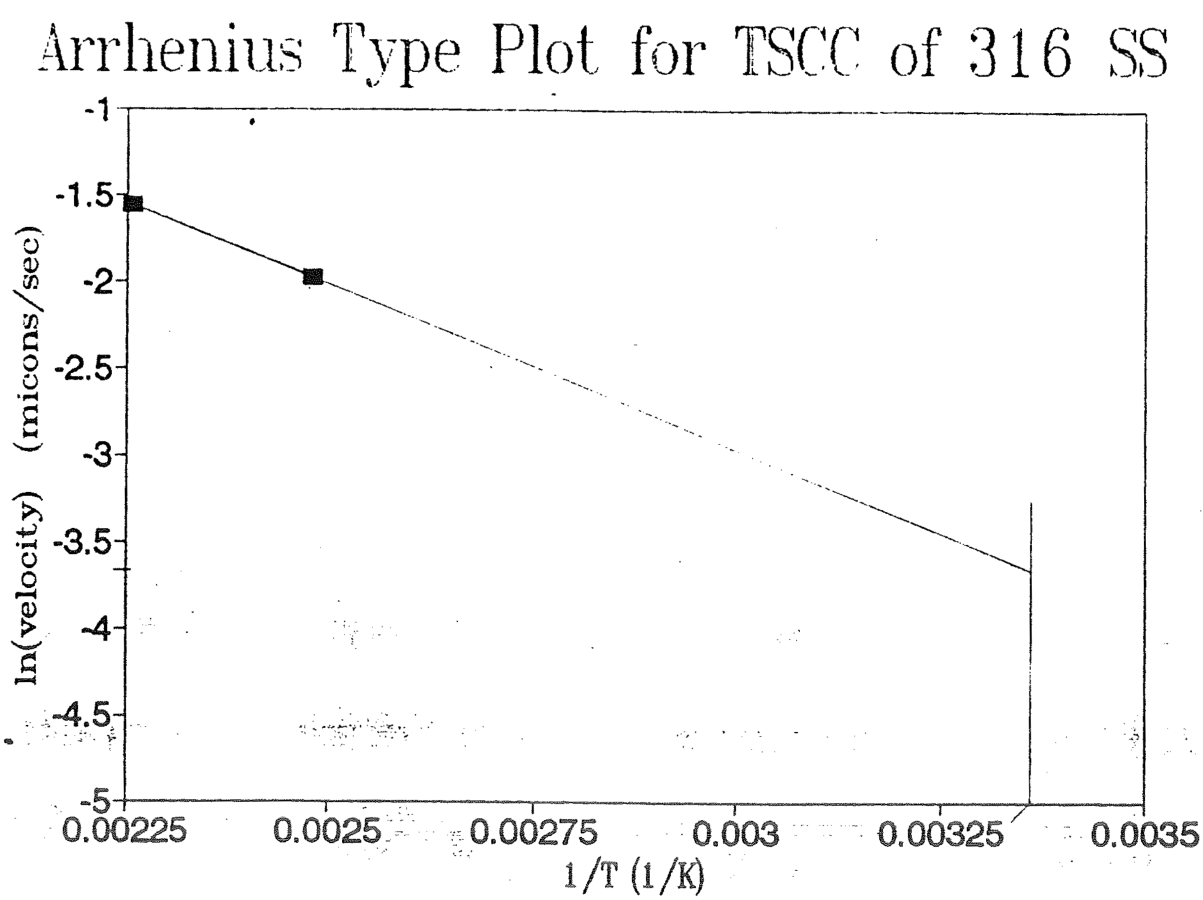


Figure 4

Task II B SCC STUDIES THROUGH LASER HOLOGRAPHY AND SPECKLE INTERFEROMETRY.

Approach

1. Double cantilever beam (DCB) Cu-30Zn brass (as shown in figure 5) are wedge loaded in 15N ammoniacal solution and placed on optical table.

2. The SCC initiation, propagation and arrest are monitored in the schematic shown in figure 6.

Progress

1. Chosen alloy and specimen geometry was shown to crack easily at room temperature in ammoniacal solution with short initiation time and appreciable propagation rate to allow real-time holography.

2. Real-time holography is demonstrated to be successful in determining SCC initiation time. Figure 7 shows a typical holograph with fringe patterns surrounding the crack tip. In many alloys initiation time is the predominant parameter which governs the extent of SCC susceptibility.

3. Local displacements due to SCC have been successfully measured. Finite element model has been developed to verify measured displacement data and determine stress field. A typical result is represented in figure 8.

4. The experimental procedure to evaluate K_{ISCC} is being finalized.

5. Work is continuing to measure SCC crack velocity by speckle photography.

6. A modified specimen geometry and experimental set up is developed for speckle photography as shown in figures 9 and 10.

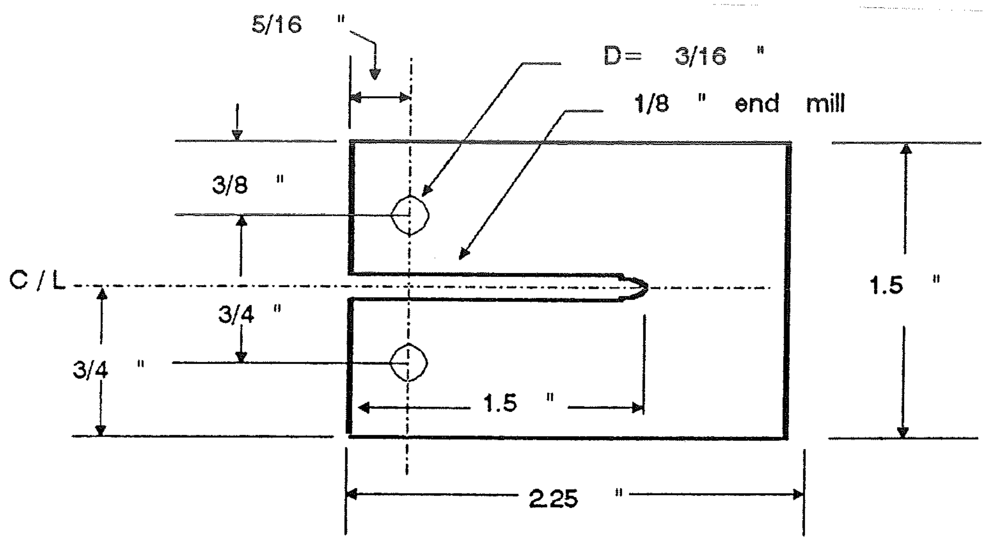


Figure 5 DCB specimen geometry

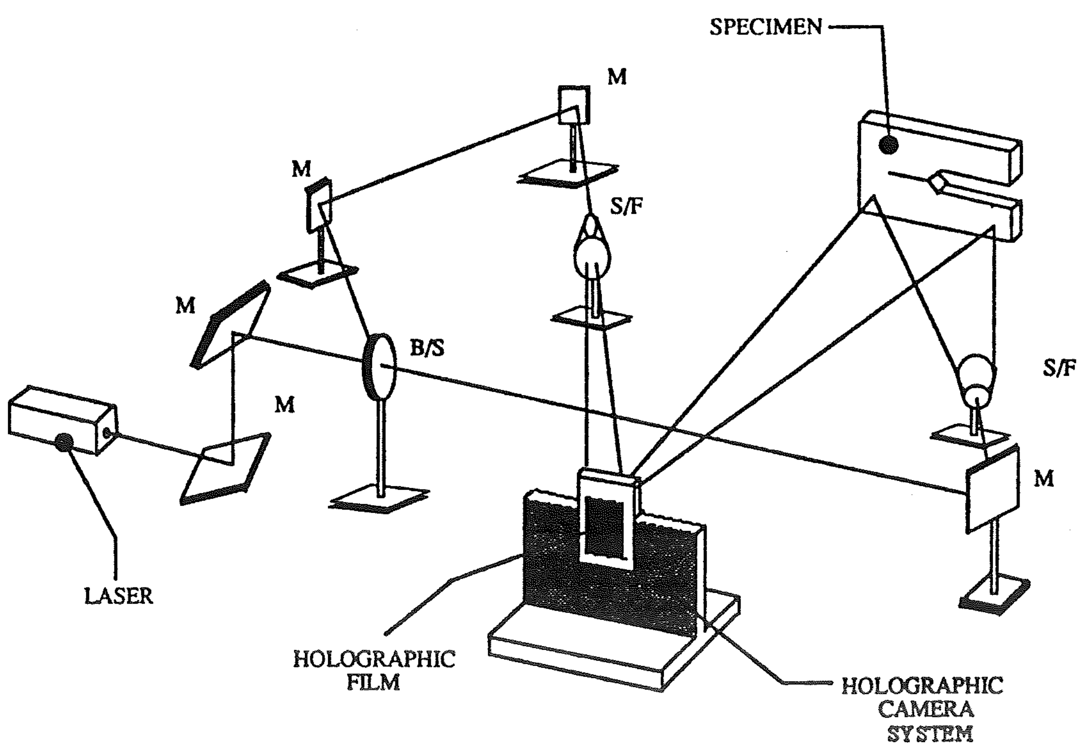


Figure 6 Coherent optics test setup

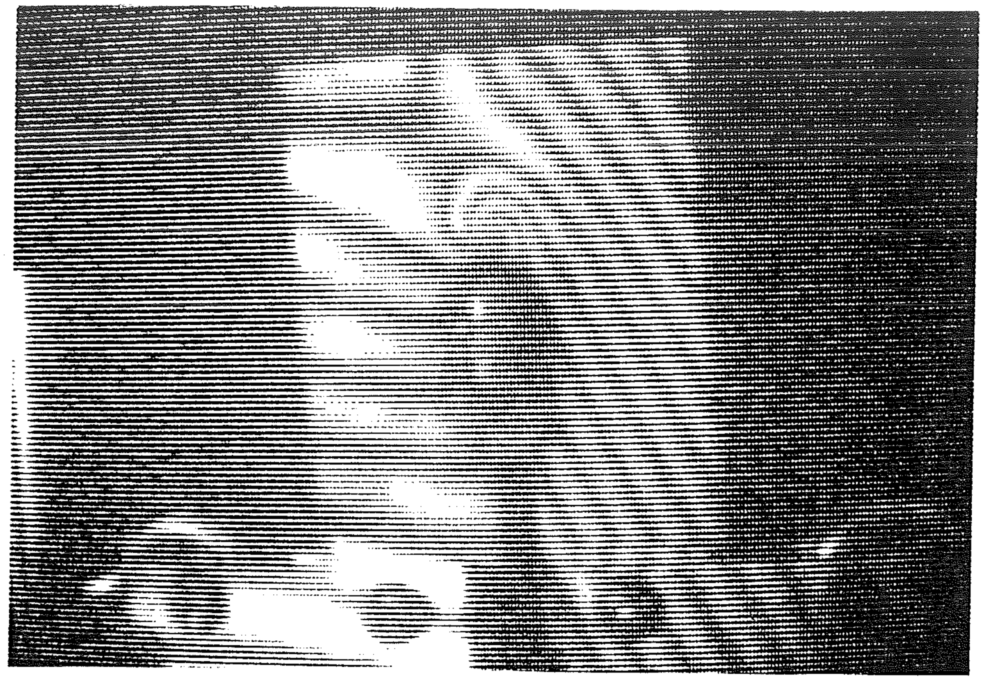


Figure 7 Holograph with fringe patterns surrounding crack tip

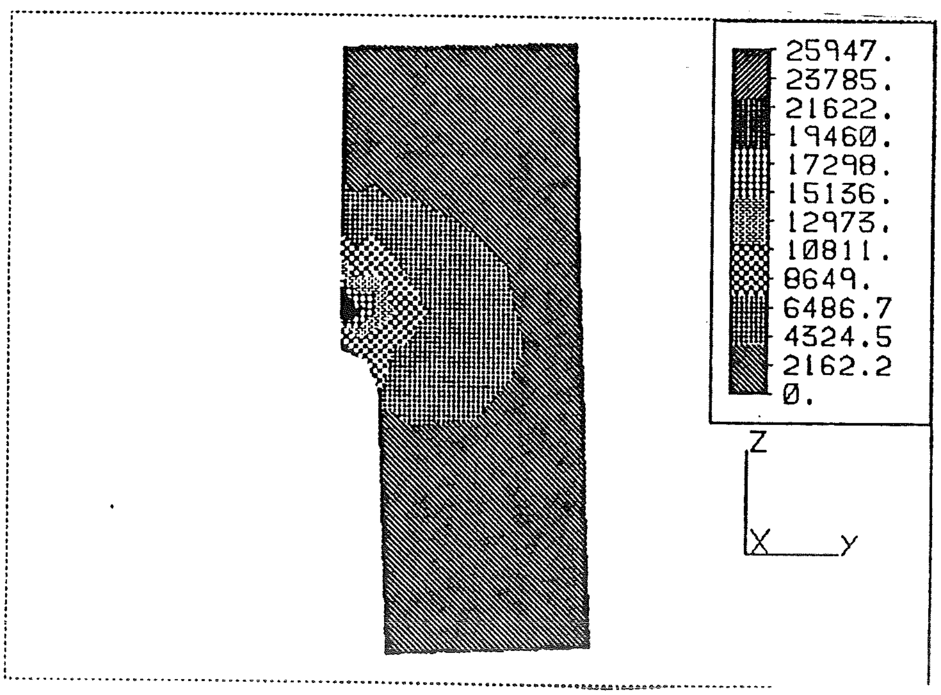


Figure 8 Stress field calculation

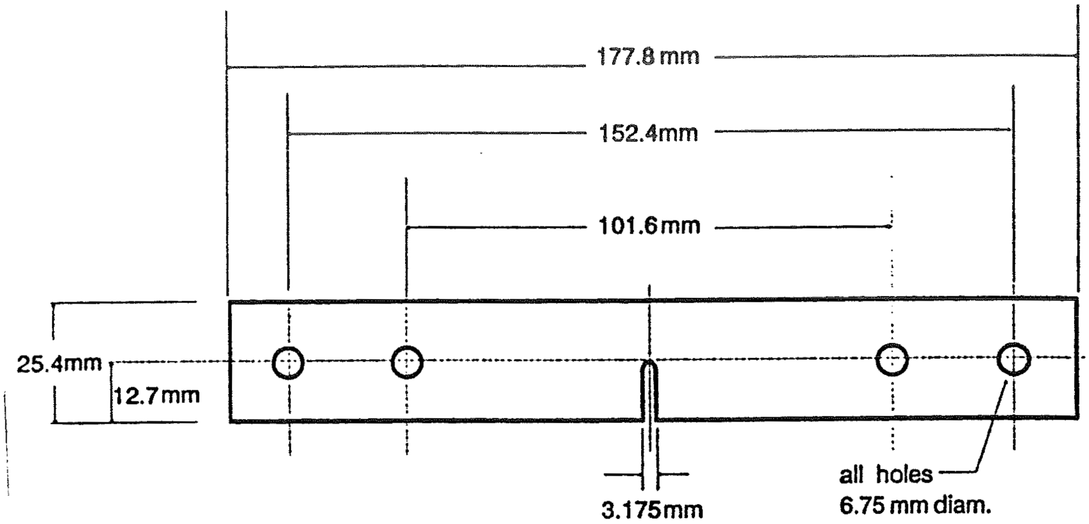


Figure 9 Modified specimen geometry

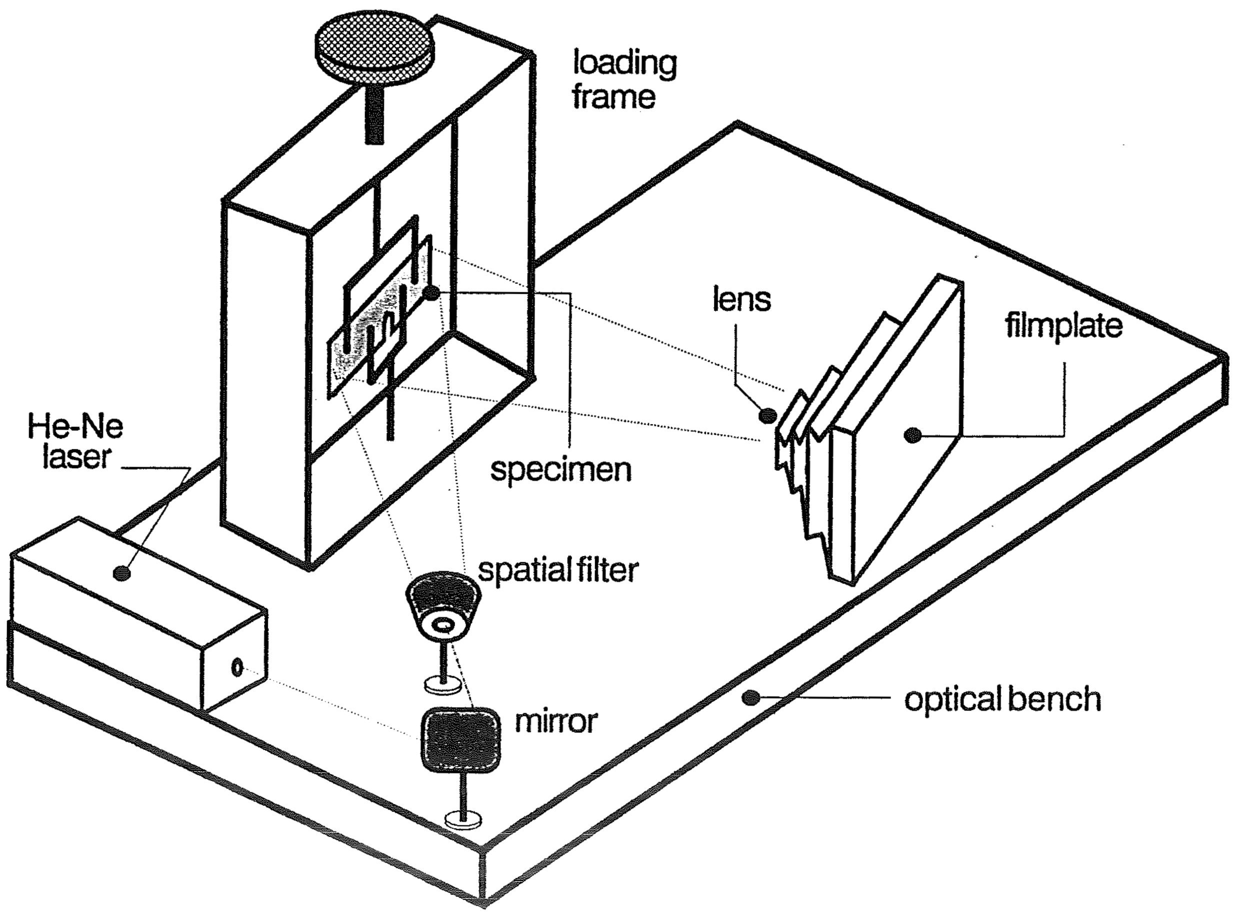


Figure 10 Test setup for speckle photography

Task III. HYDROGEN SUSCEPTIBILITY STUDIES.

Approach

1. This study is divided into two tasks. The first task is to measure coefficient of diffusion of hydrogen in Hastelloys C-22, C-276, G30 and Inconels 625 and 718.

2. The diffusivity is measured using Devanathan-Stacharsky method in which hydrogen in cathodically charged on one side of the thin foil specimen and is oxidized on the other side as it diffuses through. The steady state permeation current is related to diffusivity by one of the several mathematical techniques. This experimental setup is shown in figure 11.

3. The second task is to assess the hydrogen embrittlement susceptibility of the above mentioned materials. This will be accomplished by cathodically charging cylindrical tensile test-type specimens to hydrogen saturation followed by SSR tests.

4. The hydrogen susceptibility will be determined by the loss of strength, loss of ductility and fractographic evaluation.

5. The fractographic evaluation will include determination of fracture area ratio (FAR), dimple size ratio (R), and scanning auger microscopy (SAM) to study effect of cold working and aging on grain boundary precipitation or impurity segregation.

Progress

1. Thin foil specimens for H permeation studies and cylindrical specimens for H embrittlement studies have been procured or fabricated for all the alloys listed above.

2. Established the methodology of H permeation tests. Verified the operation and performance of the permeation cell by testing annealed iron. Iron has a very high diffusivity even at room temperature with $D = 2.7 \times 10^{-6}\ cm^2/s$. The permeation record for annealed Fe is shown in figure 12. Table IV shows the values of diffusion coefficient obtained by three different analysis. They all seem to be in reasonable agreement with the expected value of H diffusivity.

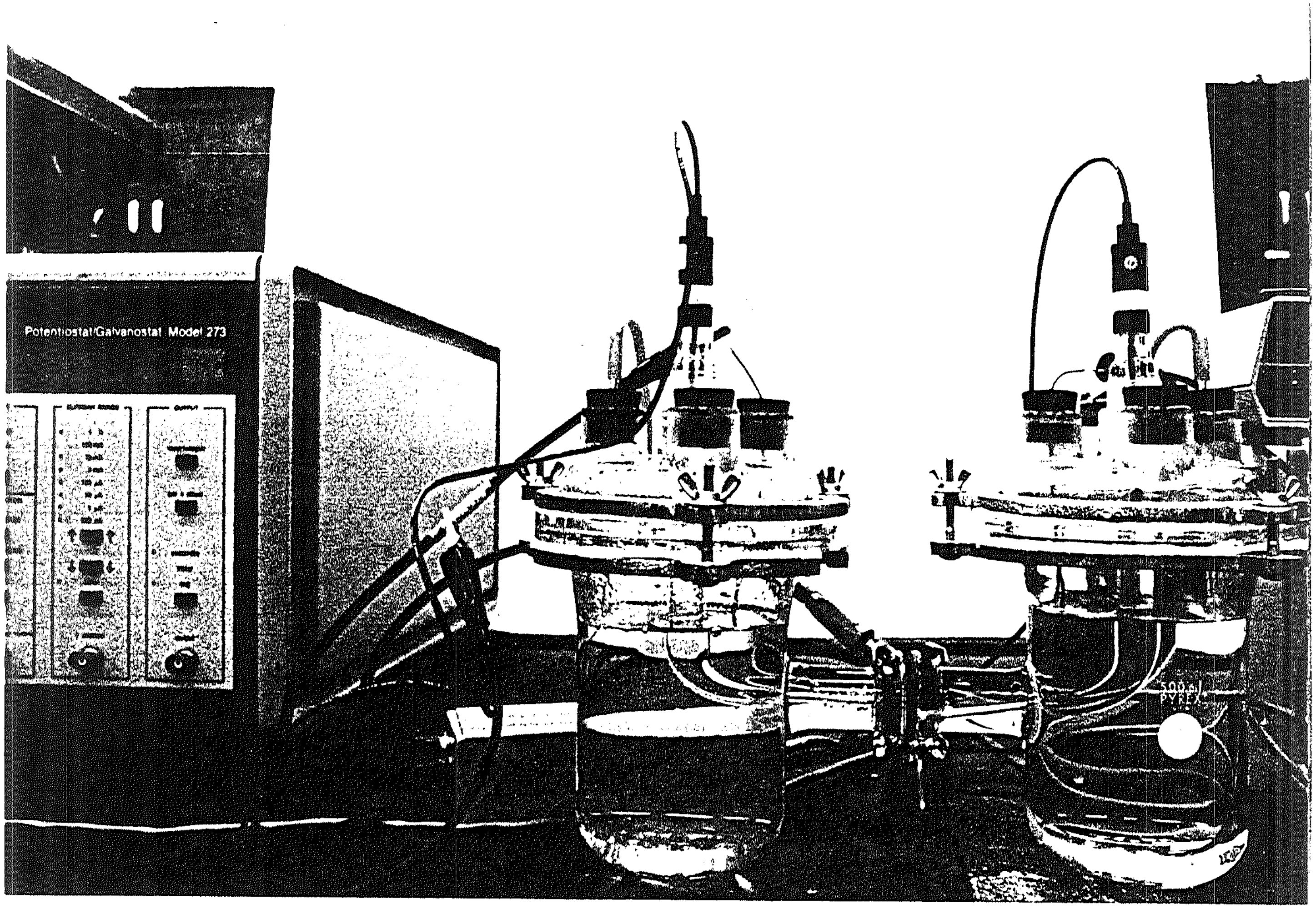


Figure 11 Experimental setup for H permeation studies

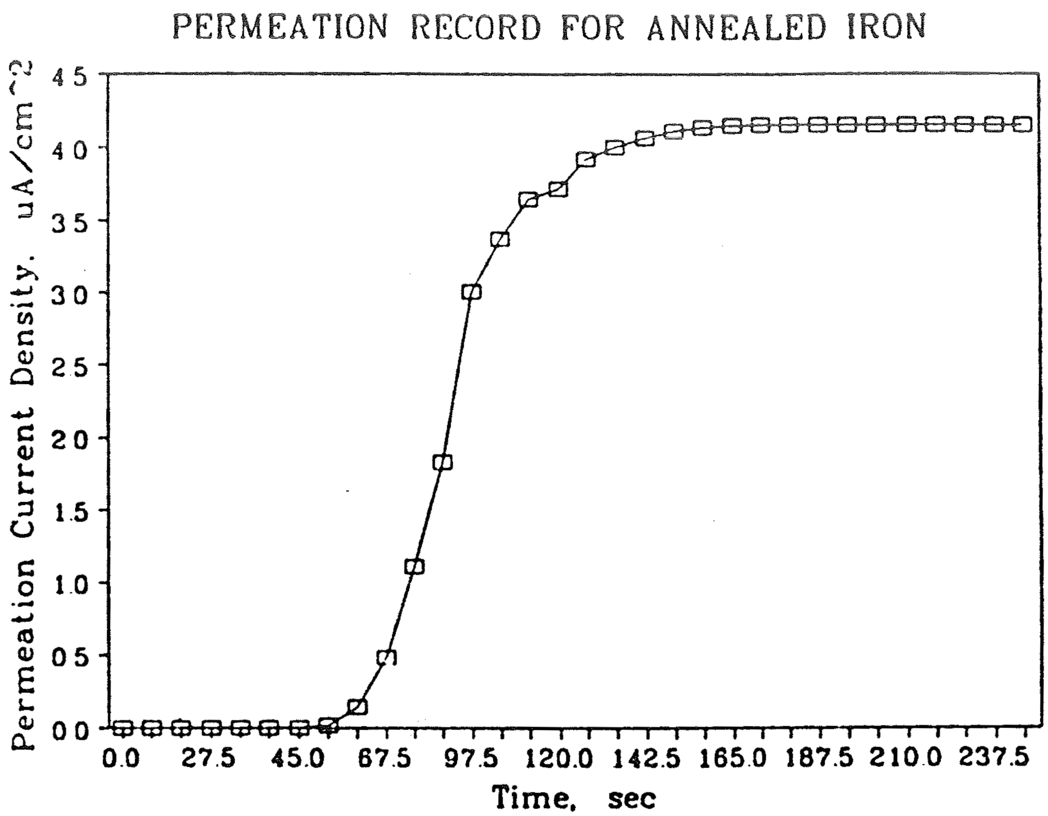


Figure 12

Results of Permeation on Annealed Iron

Diffusion Coefficient by Lag Time Calculation

$\underline{9.94 \times 10^{-7}\ cm^2/s}$

Diffusion Coefficient by Breakthrough Calculation

$\underline{7.93 \times 10^{-7}\ cm^2/s}$

Diffusion Coefficient by Rise Time Calculation

$\underline{2.37 \times 10^{-6}\ cm^2/s}$

Table IV

3. Problems encountered in the experimental setup as reported in the quarterly presentations have been satisfactorily corrected.

4. With regards to embrittlement studies, Inconel 625 specimen has been tested in air by slow strain rate pulling at 10^{-6} per second. Another specimen of the same material is being cathodically charged by hydrogen before performing SSR tests.

5. Preliminary SEM and SAM fractographic studies have been performed to assess the impurity segregation to grain boundaries.

6. This research is being coordinated with White Sands Lab who will use hydrogen gas pressure to rupture the specimen discs and assess embrittlement.

S7-81
333527

N91-70705

UNIVERSITY OF CENTRAL FLORIDA

COLLEGE OF ENGINEERING

ORLANDO, FLORIDA 32816

ANNUAL PROJECT REPORT

to

NASA KENNEDY SPACE CENTER

NASA -- PROJECT MANAGEMENT

16-10-204

under

COOPERATIVE AGREEMENT NCC-10-0003

for calendar year 1990

Stephen L. Rice, PhD, PE
Professor and Associate Dean
Principal Investigator

Background

In the fall, 1988, exploratory meetings were held to determine ways in which the College of Engineering (COE) of the University of Central Florida (UCF) could assist Kennedy Spaceflight Center (KSC) in its mission oriented activities. In these meetings, NASA personnel briefed COE faculty and administrators on programmatic needs, and COE faculty presented their professional interests and expertise. Based upon these exchanges, it became evident that a number of projects could be conducted by UCF faculty in various colleges in support of KSC needs and mission objectives.

In March, 1989, UCF/COE submitted a proposal to KSC entitled "Cooperative Agreement for Technology Development -- Task Definition, Summer, 1989". The purpose of this work was to "enable improved problem definition and to initiate working relationships between UCF and KSC". Several specific work areas were identified, including Shuttle Close-Out Data Systems (Dr. Ragusa), Public Affairs Information Retrieval (Dr. Myler and Dr. Driscoll), Knowledge Acquisition from Natural Language Input (Dr. Gomez), Non-Destructive Evaluation (Dr. Moslehy), Corrosion Protection (Dr. Desai), and Shuttle Tile Processing (Dr. Hosni). A second proposal followed in June, 1989, to extend the process of defining specific tasks which would form the basis of a Cooperative Agreement covering several areas of technology. These were funded under Grant NAG-10-0058, which enabled work to continue into the beginning of 1990, when year 1 of the Cooperative Agreement NCC-10-0003 began.

Work Effort

Following the foundation laid under project NAG-10-0058, the Project Management activities for the Cooperative Agreement included several elements. First among these was coordination between the various academic units (Colleges of Engineering, Arts and Science, Business Administration, Education; and Departments of Computer Science, Mechanical and Aerospace Engineering, Industrial Engineering and Management Sciences, Computer Engineering, Management). Separate projects were conducted under eight separate UCF account numbers (plus one additional control account), each with one Principal Investigator and, depending upon the project, various students (graduate and undergraduate) and Co-Principal Investigators.

Regular Project Review Meetings were held on a quarterly basis, with the venue alternating between KSC and UCF. These meetings allowed for presentation of work accomplished to date, for suggestions on future activities, and for interaction between UCF and KSC personnel. Where appropriate, laboratory demonstrations were scheduled as a part of these meetings. It was apparent, through such regular interaction on each project, that the individual work efforts were going forward to effectively meet the needs and mission-oriented objectives of KSC.

Other activities under the Project Management include the continuing development of accounting procedures, databases, and user-friendly formats for presentation of information to PIs.

This work was accomplished by COE and Division of Sponsored Research (DSR) staff at UCF, and allowed overall monitoring and control of funding for each subproject of the Cooperative Agreement. Weekly meetings were held to review accounts and to evaluate budgetary situations, and individual PIs were contacted as necessary on various fiscal and work-related aspects of the projects.

The first year of the Cooperative Agreement ended approximately 11 December 1990, and the second year has begun with one project task having been phased out (Dr. Myler), and with two additional projects scheduled to come on-line (Dr. Gunnerson and Dr. Clausen). The ability to add and subtract projects as KSC needs dictate and UCF resources permit is an exceedingly valuable component of the Cooperative Agreement.

N91-70706

58-44

333528

AN INTERIM STATUS REPORT TO
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
PHASE I (MAY 1990 - MAY 1991)

p-12

HEAT PIPES AND PHASE-CHANGE MATERIALS FOR COMMERCIAL AND RESIDENTIAL APPLICATIONS

BY

F.S. GUNNERSON AND G.E. THORNCROFT
UNIVERSITY OF CENTRAL FLORIDA
COLLEGE OF ENGINEERING
ORLANDO, FLORIDA 32816

JANUARY 31, 1991

ABSTRACT

This report briefly summarizes the research activities at the University of Central Florida (UCF) from May 1990 through January 1991, which focused on the application of heat pipes and phase-change materials toward thermal energy storage.

Thermal energy storage (TES) has been shown to be an effective method to shift electrical demand periods by augmenting residential and commercial space heating. Research efforts specifically addressed storing thermal energy in solid state phase change materials (PCMs) and delivering this heat by a number of methods, most notably by heat pipes. Also incorporated into this research project were the efforts of two senior-level mechanical engineering student design teams from EML 4304 (Measurement Laboratory, 3 students) and EML 4501 (Senior design, 6 students).

Research in PCMs and heat pipes has produced four TES module designs, which are summarized within the report. Each design incorporates a solid state PCM called pentaerythritol, contained by various methods within an electrically heated storage tank. The heat contained within the tank is transported either directly to the living space or indirectly via the existing HVAC ductwork. Methods to deliver the heat to the duct include circulating a heat transfer fluid and embedding heat pipes into the PCM.

SUMMARY OF ACTIVITIES
(MAY 1990 - JANUARY 1991)

Electrically-charged thermal energy storage is a viable technique for utility companies to reduce the peak demand for electric power. As part of NASA's Technology Utilization program, the University of Central Florida (UCF) has been performing research on heat pipes and phase-change materials to be applied to the design of commercial and residential TES modules. The research project thus far has been conducted during an eight month period (May 1990 through January 1991) and incorporates the efforts of one graduate and nine senior-level undergraduate students of mechanical engineering under the direction of Professor Fred S. Gunnerson. A comprehensive final project report will be issued in late May, 1991.

Research activities have focused on three areas: (1) quantifying the heat transfer characteristics of solid-state phase change materials, (2) analyzing the thermal behavior of heat pipes, and (3) using the results of this research to develop conceptual designs for TES modules. The modules are designed to be electrically charged during periods of low power demand, and then release the stored heat during a 4 to 6 hour period of high or peak power requirements. For typical residential heating, this requires storage capabilities in the 80,000 to 100,000 Btu (23 to 30 kW-hr) range. The units are designed to be compact, unobtrusive, and capable of being retrofitted near or within existing structures.

Pentaerythritol (PE), a solid-state PCM, was chosen for the storage medium for the module, and a computer model was developed to examine its heat transfer characteristics. The program consists of a transient, one-dimensional finite-differencing scheme applied to long cylinder. It includes the effects of multimode heat transfer (free and forced convection, radiation) and the variance of properties with temperature. Research continues in this area, and is being submitted to the 1991 Intersociety Energy Conversion Engineering Conference under the title, "Performance Evaluation of a Solid State Phase Change Material for Thermal Storage Applications" (a copy of the abstract is attached).

Research in heat pipes revolved around a technique to visually document and analytically model the performance characteristics of simple heat pipes. Copper heat pipes with annular copper mesh wicks and charged with refrigerant-12 were externally coated with Thermochromic Liquid Crystal (TLC) paint. The thermally sensitive TLC coating reversibly changes color upon heating, revealing temperature profiles on the surface of the pipe. A brief video documentation of the experiment was developed, which illustrated the isothermal behavior of the evaporator and condenser regions, as well as the effects of non-condensible gases present within the pipe. This research has also been submitted to the IECEC, under the title, "Visualizing the Thermal Performance of Heat Pipes with Thermochromic Liquid Crystals." (a copy of the abstract is attached).

Four engineering designs have been distilled from the conceptual design efforts; each incorporates the solid-to-solid phase transition thermal storage material, pentaerythritol (PE). Two of the designs include controllable heat pipes or thermosyphons to transfer the heat from the TES module to the residential HVAC duct. The remaining two designs utilize convective liquid or air for the heat transport from the module to the duct or space to be heated. All of the designed TES modules are electrically chargeable (6 to 8 kW) within a three to five hour period and initially store heat at 400 °F (480 °F maximum).

A brief description of each TES module follows:

(a) TES-1 This design utilizes encapsulated PE -- "canned" in standard twelve ounce beverage containers (244 cans). The PCM cans are immersed in Paratherm NF, a non-toxic, high temperature heat transfer fluid, to produce a unit with an approximate 50 gallon volume. The stored heat is transported to an in-duct liquid-to-air heat exchanger by pumping the fluid. The rate of heat transfer to the air duct is regulated by varying the speed of the pump and, correspondingly, the flow rate of the working fluid. Schematics of the TES-1 design are shown in Figures 1 and 2.

(b) TES-2 The second design utilizes encapsulated or bulk PE for energy storage with Paratherm NF as an in-tank heat transfer medium. The heat is removed from the thermal energy storage tank (approximately 55 gallon) by a novel controllable heat pipe and transported to an in-duct heat exchanger which serves as the condenser to the heat pipe system. The rate of heat transfer to the air duct is controlled by regulating the condensate return to the heat pipe. This design is inherently simple but must be located adjacent to and below the air duct system. This design is conceptually illustrated in Figure 3.

(c) TES-3 The third design is an "oven-type" unit which utilizes bulk PE with embedded conventional heat pipes or thermal syphons. Heat is removed directly from the PE and transferred to the air duct by finned heat pipes, and thus, like TES-2, does not require a separate in-duct heat exchanger. Control of the unit is facilitated by mechanically adjusting a heat pipe/PE cover and/or in-duct air mixing. This unit must be positioned on-line with the supply air duct. Total TES module volume is approximately 35 gallons without the air control portion of the unit. Figure 4 conceptually illustrates the TES design.

(d) TES-4 This is the simplest, least expensive, and most versatile design. Like TES-1, canned PE is stacked within an electric "oven-type" geometry. All heat exchange is by natural or forced air convection. There is no liquid heat transfer medium. During discharge, air is drawn inside from, and discharged to, the domestic space. The rate of air heating is regulated by dampers or louvers on the unit. The design can be located and sized for a range of local space heating requirements. Figure 5 depicts the design.

FUTURE RESEARCH ACTIVITIES
(JANUARY 1991 - MAY 1991)

Research on heat pipes and phase change materials continues on time and within budget constraints. Two additional senior design groups have been enlisted to perform research; one group is studying the effects of charge -- the amount of fluid present within a heat pipe -- on a heat pipe's performance. The other group is studying the encapsulation of pentaerythritol in metal containers, which includes chemical and thermal studies of the materials. The work of these groups will be applied directly to the development of a working prototype of the Thermal Energy Storage module, which will continue in Phase II.

A detailed final report for Phase I will be submitted in May. It will include copies of all student research reports submitted during the period. Also included in the report will be an outline of research that will continue in Phase II.

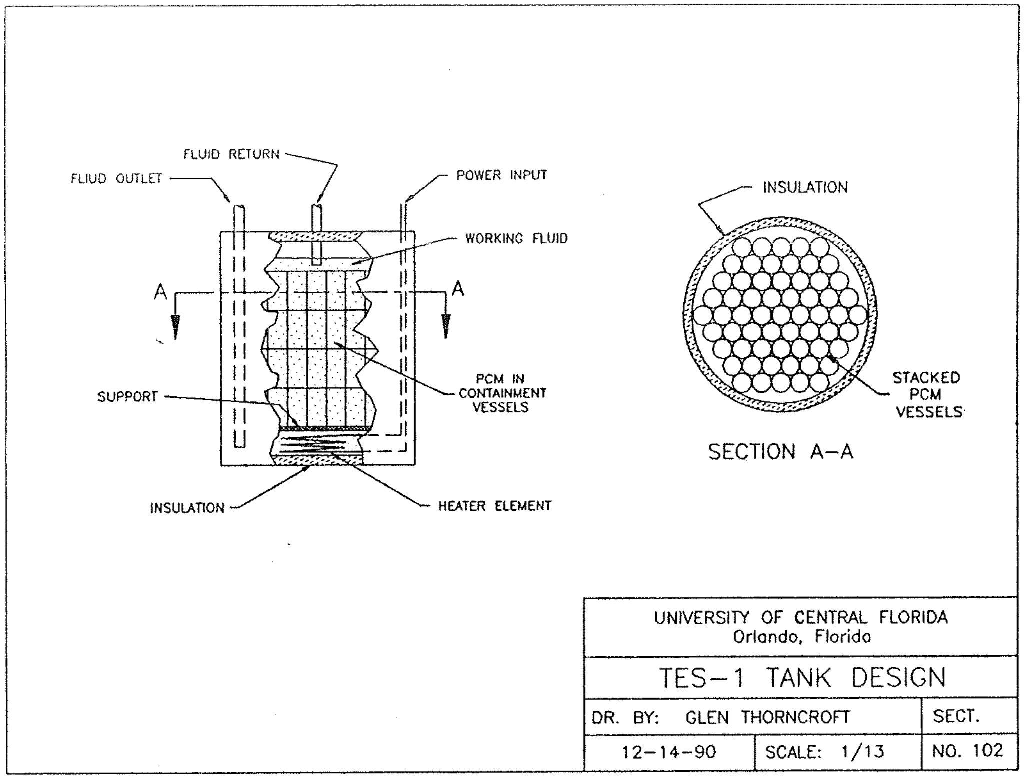


FIGURE 1 PCM STORAGE TANK FOR TES-1

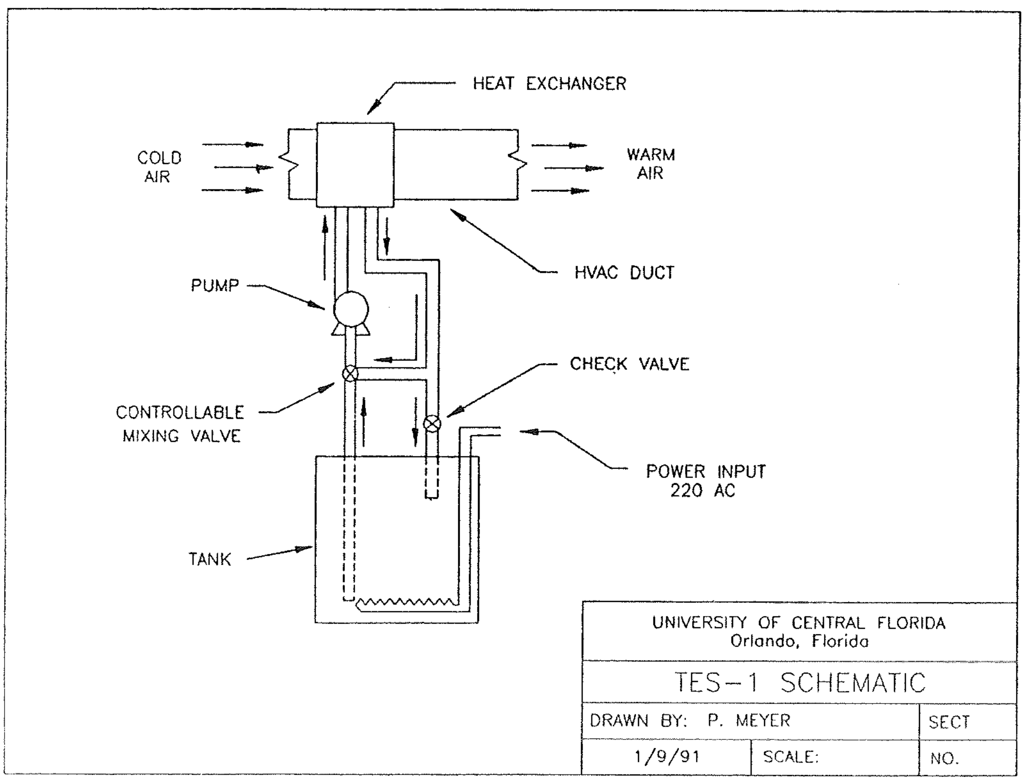


FIGURE 2 SCHEMATIC DIAGRAM FOR TES-1

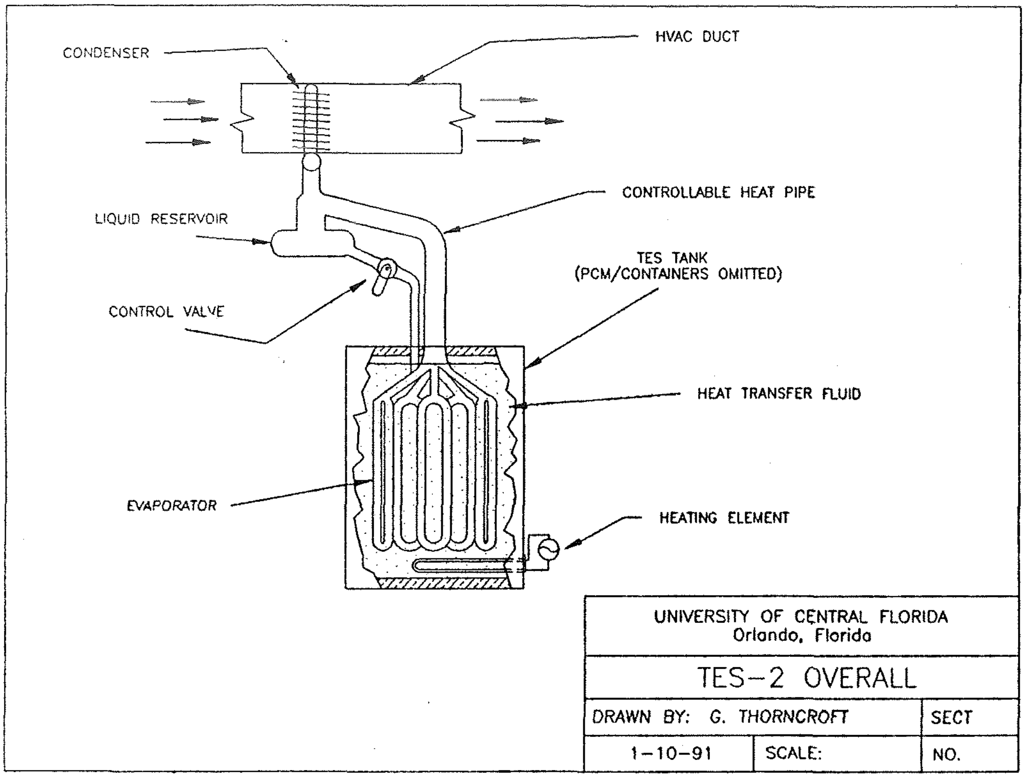


FIGURE 3 OVERVIEW OF DESIGN TES-2

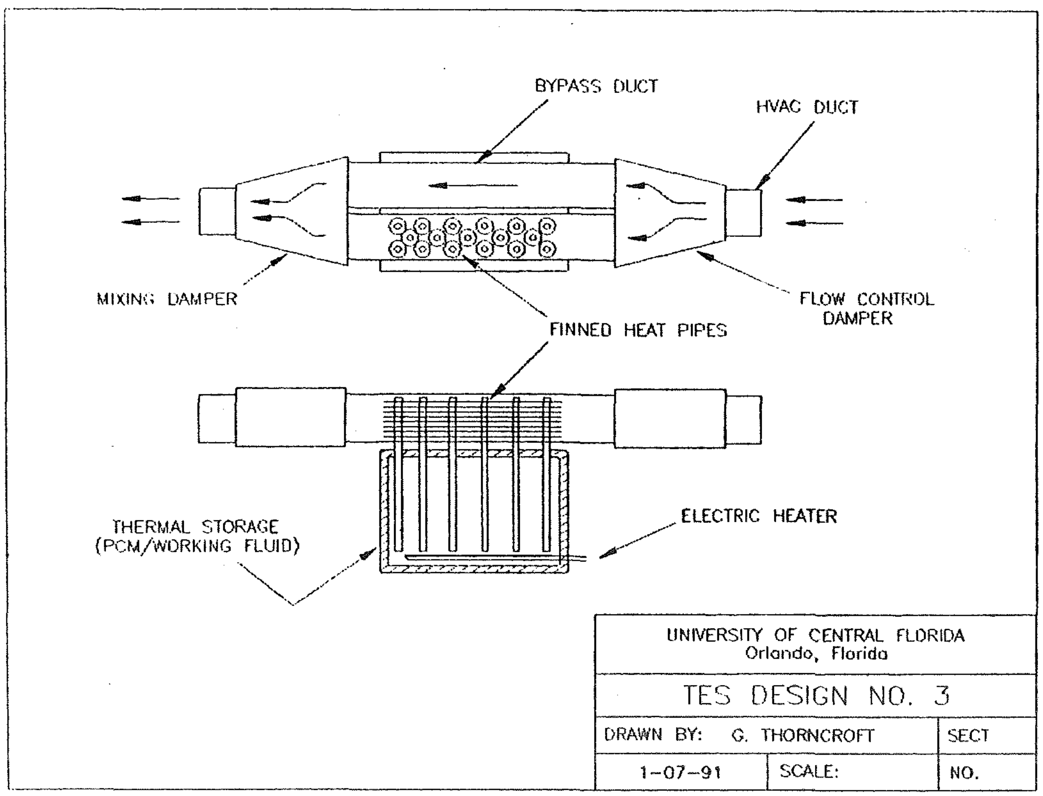


FIGURE 4 CONCEPTUAL ILLUSTRATION OF TES-3

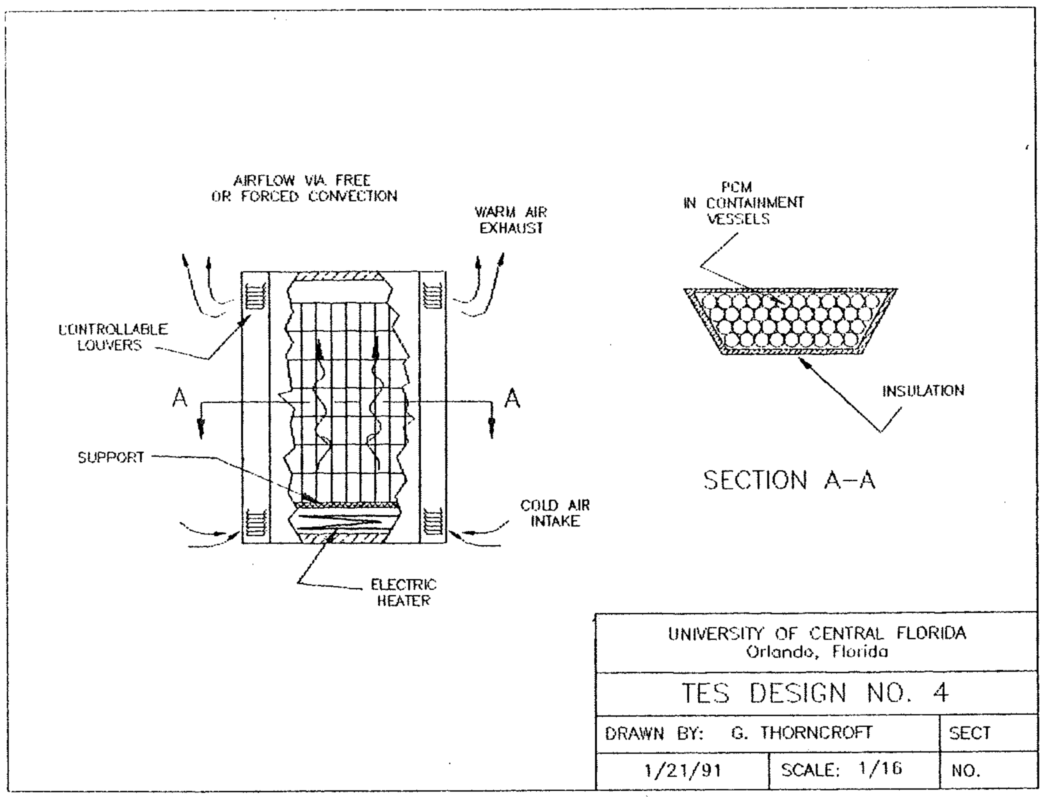


FIGURE 5 CONCEPTUAL DESIGN FOR TES-4

SUBMITTED FOR: 26th Annual IECEC, Boston, August 4-9, 1991
TOPICAL AREAS: Heat Pipes, Aerospace Power Systems, Conversion Technologies, Thermal Management

Visualizing The Thermal Performance of Heat Pipes with Thermochromic Liquid Crystals

By

Fred S. Gunnerson and Glen E. Thorncroft
University of Central Florida
Mechanical and Aerospace Engineering
Orlando, FL 32816

ABSTRACT

A novel and instructive program has been conducted to visually document and analytically model the thermal performance characteristics of simple heat pipes. Copper heat pipes (15mm x 730mm) with annular copper mesh wicks and charged with refrigerant-12 were externally coated with thermochromic liquid crystal (TLC) paints. The thermally sensitive TLC coating reversibly changes color upon heating and readily permits visual identification of transient and steady state isotherms during low temperature (approximately 30 to 60 degrees C) operation (1). The startup and operational behaviors of the heat pipes as well as the presence of non-condensible gases can be visually identified through a spectrum of color changes; each color indicative of an isothermal region as indicated on Figure 1. With heat pipe surface temperatures visually quantifiable, analytical modeling of the heat transfer characteristics can be better defined and verified.

A brief video documentation of heat pipe thermal performance, using the TLC technique, is presented. In addition, the visually quantifiable temperature characteristics are shown to support and better refine analytical modeling efforts. Specifically, the following heat pipe performance characteristics are illustrated and discussed:

(i) A side-by-side comparison of the transient heatup of a solid copper rod and an equivalent heat pipe dramatically illustrates the heat pipe's very high effective thermal conductance and the "equivalent thermal conductivity" principle (2).

(ii) The isothermal, low thermal impedance nature of an operating heat pipe is shown.

(iii) The migration and collection of injected non-condensible gases within the condenser region of the heat pipe is apparent.

(iv) The TLC technique can be used to better quantify and verify analytical heat transfer modeling efforts.

The experimental and analytical results presented were developed, in part, from a thermal energy storage program incorporating heat pipes and solid-to-solid phase transition materials. The program is supported by the NASA Office of Technology Utilization and the Florida Power Corporation.

1. M. Parsley, "An Introduction to Thermochromic Liquid Crystal Products", Hallcrest Co., Glenview, Illinois, November 1987.

2. P. D. Dunn and D. A. Reay, Heat Pipes, Third Edition, Pergamon Press, 1982.

3. R. J. Moffat, "A Short Course in Temperature Measurement" for the 36th International Instrumentation Symposium of the Instrumentation Society of America, Denver, Colorado, May 6-7, 1990.

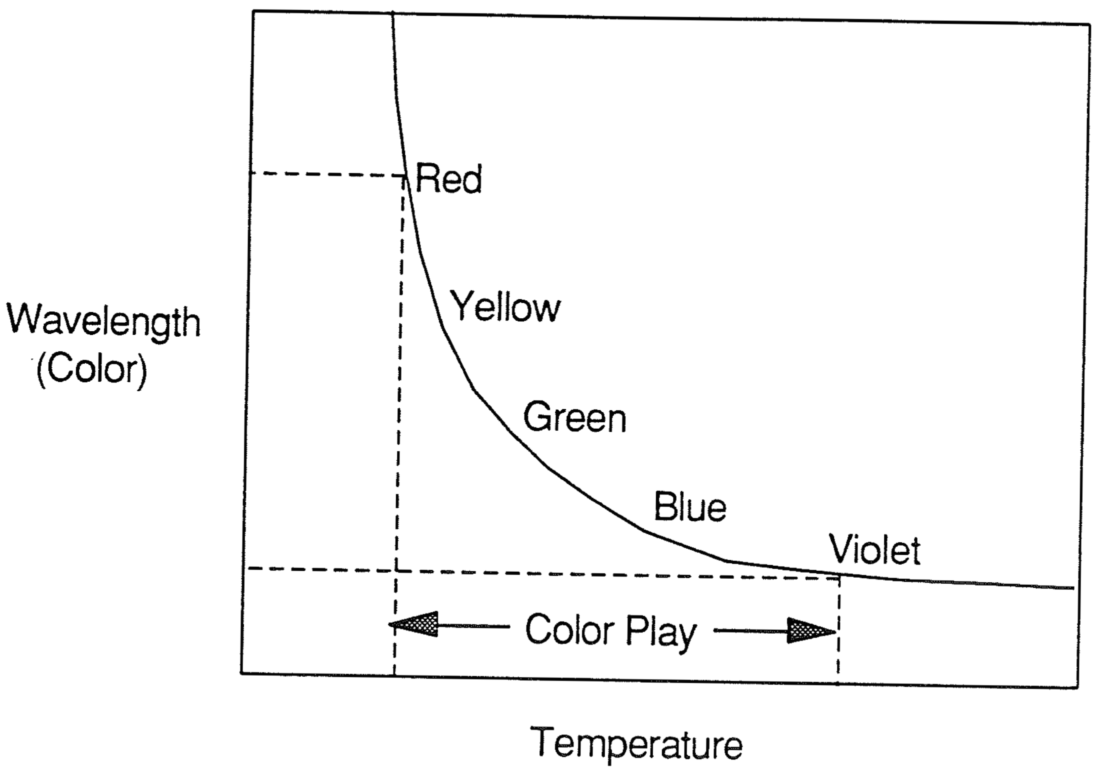


FIGURE 1 QUALITATIVE LIQUID CRYSTAL RESPONSE (1,3)

SUBMITTED FOR: 26th Annual IECEC, Boston, August 4-9, 1991

TOPICAL AREAS: Thermal Energy Storage, Thermal Management

PERFORMANCE EVALUATION OF A SOLID STATE PHASE CHANGE MATERIAL FOR THERMAL STORAGE APPLICATIONS

By

Glen E. Thorncroft and Fred S. Gunnerson
University of Central Florida
Mechanical and Aerospace Engineering
Orlando, Florida 32816

ABSTRACT

Phase change materials are becoming increasingly attractive as thermal energy storage (TES) media. Most phase change materials (PCMs) used for this purpose undergo a solid-to-liquid transition (melting) which reversibly stores large quantities of heat. However, some materials, such as pentaerythritol and trimethylolethane [1], undergo a solid-to-solid transition, and hence retain their shape throughout their use. This makes solid state PCMs more versatile, since they can be cast in a variety of shapes and their need for containment is reduced [2].

Efforts to model phase changes have generally been applied to more common solid-liquid PCMs. In analyses of these materials, an approximation is often made that convection in the liquid phase is not present; that is, that the liquid phase behaves essentially as a solid. In reality, convective effects in the liquid phase can greatly affect the heat transfer characteristics of the material. However, in spite of potentially large errors, this simplification is often essential to solve the problem.

For a solid-state PCM, the second phase is indeed solid, so the above simplification is no longer an approximation. As a result, a fundamental error is removed from the phase-change model. The behavior of solid-state PCMs can thus serve to test the validity of other approximations whose effects on the model may have previously been too small to quantify. The phase change model itself can then be better examined and refined.

A one-dimensional transient computer model was developed to determine the temperatures within a material undergoing solid state phase change. The purpose of this model was to predict the heat transfer characteristics of pentaerythritol cast in simple geometries and exposed to several modes of heat transfer. The research was performed in conjunction with the development of a

thermal energy storage device being developed for residential heating. The project is funded by NASA, Office of Technology Utilization and the Florida Power Corporation.

The computer program consists of a time-dependant finite-differencing scheme which presumes an isothermal phase change at each node. Heat transfer to or from the solid is by means of radiation and natural or forced convection. Variance in material properties with temperature and phase are included. If a material, such as copper or aluminum, is used to contain the PCM, it can also be accounted for by appropriate nodes in the grid [3].

Tests were performed to verify the program, and to examine the effect of dopants on enhancing the conductivity of pentaerythritol. A sensitivity analysis was performed on the model to evaluate the role of thermal properties on phase change influenced heat transfer. Test procedures and results are presented and compared with theoretical predictions.

REFERENCES

1. D.K. Benson, et al., "Solid State Phase Transitions in Pentaerythritol and Related Polyhedric Alcohols," Solar Energy Materials, Vol. 13, North-Holland, 1986.

2. D.V. Hale, et al., "Phase Change Materials Handbook," NASA CR-61363, 1971.

3. G.E. Meyers, Analytical Methods in Conduction Heat Transfer, McGraw-Hill, 1971.